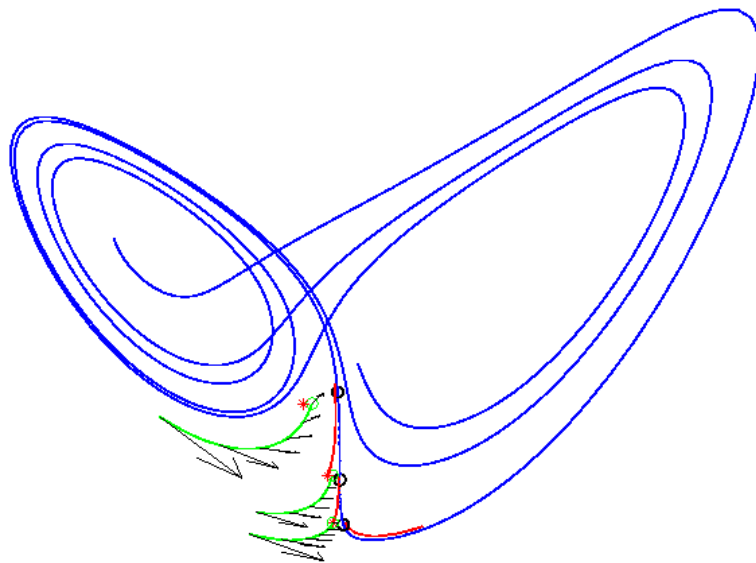


Dottorato di Ricerca in Modellistica Fisica per la Protezione dell'Ambiente – XX Ciclo  
Dipartimento di Scienze della Terra e Geologico-Ambientali – Dipartimento di Fisica  
Settore Scientifico-Disciplinare FIS/06 – Fisica per il Sistema Terra e il Mezzo Circumterrestre  
COORDINATORE: *Prof. Ezio Todini*

# A Dynamical System Approach to Data Assimilation in Chaotic Models



DOTTORANDO:  
*Dott. Massimo Pilolli*

RELATORI:  
*Prof. Rolando Rizzi*  
*Dott.ssa Anna Trevisan*



*A mio padre e mia madre*





“Une cause très petite, qui nous échappe, détermine un effet considérable que nous ne pouvons pas ne pas voir, et alors nous disons que cet effet est dû au hasard. Si nous connaissions exactement les lois de la nature et la situation de l’univers à l’instant initial, nous pourrions prédire exactement la situation de ce même univers à un instant ultérieur. Mais, lors même que les lois naturelles n’auraient plus de secret pour nous, nous ne pourrions connaître la situation initiale qu’*approximativement*. Si cela nous permet de prévoir la situation ultérieure *avec la même approximation*, c’est tout ce qu’il nous faut, nous disons que le phénomène a été prévu, qu’il est régi par des lois; mais il n’en est pas toujours ainsi, il peut arriver que de petites différences dans les conditions initiales en engendrent de très grandes dans les phénomènes finaux; une petite erreur sur les premières produirait une erreur énorme sur les derniers. La prédiction devient impossible et nous avons le phénomène fortuit.”

Jules-Henri Poincaré – Science et Méthode, 1908



# Acknowledgements

First of all, I wish to write a note in memory of Edward Norton Lorenz, who died in Cambridge, MA, USA, on April 16. He was not only the father of Chaos Theory and the butterfly effect, one of the great scientific revolutions of the past century, but also the grandfather of all meteorologists. I regret I have never had the chance to meet him.

I wish to thank my advisor Anna Trevisan (ISAC-CNR), one of the strongest and most motivated woman I ever met: her deep insight into data assimilation, her constant support and enlightening discussions made this work possible. A special thank you to Alberto Carrassi and Francesco Uboldi for their frequent, friendly help. I also wish to thank Prof. Rolando Rizzi and Prof. Ezio Todini for their support, useful suggestions and comments.

Truly unforgettable have been the kind hospitality and the useful discussions with Prof. Catherine Rouvas-Nicolis and Dr. Stéphane Vannitsem, at the Department of Meteorological Research and Development of the *Institut Royal Météorologique/Koninklijk Meteorologisch Instituut*, the Royal Meteorological Institute of Belgium, Brussels, where I met very nice and friendly people: *merci beaucoup/dank u wel!*

E infine un grazie grande così alla Stefania, per la sua pazienza e disponibilità!

The software used in this work has been developed on the basis of software written in the *NERC Data Assimilation Research Centre*, Department of Meteorology, University of Reading, United Kingdom. All experiments were run on a *Debian 3.1* Linux computer, using both *Matlab* 6.5.0.180913a Release 13 and *gcc/g++* 3.3.5 C/C++ compiler. Some figures were produced by *gnuplot* 4.0.

*M. P., Bologna, April 2008*



# Contents

|   |             |
|---|-------------|
| <b>Acknowledgements</b>   | <b>v</b>    |
| <b>List of Figures</b>  | <b>xi</b>   |
| <b>List of Tables</b>   | <b>xv</b>   |
| <b>Abbreviations</b>  | <b>xvii</b> |
| <b>Partial List of Symbols</b>  | <b>xix</b>  |
| <br>  |             |
| <b>1 Introduction</b>   | <b>1</b>    |
| 1.1 A historical perspective . . . . .  | 1           |
| 1.2 Data Assimilation . . . . .   | 4           |
| 1.3 Framing the problem . . . . .   | 5           |
| 1.3.1 Basic features of chaotic systems, a historical note . . . . .            | 6           |
| 1.3.2 Conservative, nonconservative dynamical systems . . . . .                 | 6           |
| 1.3.3 Probability Density Functions and Liouville equation . . . . .            | 7           |
| 1.3.4 The Markov processes and the Fokker-Planck equation . . . . .             | 8           |
| 1.3.5 The Liouville Theorem . . . . .   | 9           |
| 1.3.6 Dissipative systems, attractors and strange attractors . . . . .          | 11          |
| 1.3.7 Small perturbations dynamics, tangent linear model, adjoint model . . . . | 12          |
| 1.3.8 Lyapunov vectors and Lyapunov exponents . . . . .                         | 13          |
| 1.3.9 Bred vectors . . . . .  | 15          |
| 1.3.10 Dynamical systems described by maps . . . . .                            | 16          |
| 1.4 Lorenz's three dimensional chaotic system (1963) . . . . .                  | 20          |
| 1.4.1 The equations . . . . .   | 20          |
| 1.4.2 The meaning of variables and parameters . . . . .                         | 22          |

|          |  |           |
|----------|--|-----------|
| 1.4.3    | Why it is so important: chaos implies limited predictability . . . . .           | 27        |
| 1.4.4    | Lyapunov exponents, dimensionality and doubling time . . . . .                   | 27        |
| 1.5      | The aim of this study . . . . .  | 28        |
| 1.6      | Notation and conventions . . . . .   | 28        |
| 1.6.1    | Model space and observational space . . . . .                                    | 29        |
| 1.6.2    | The observation operator $H$ and its linear approximation $\mathbf{H}$ . . . . . | 29        |
| 1.6.3    | Error vectors . . . . .  | 30        |
| 1.6.4    | Error covariance matrices . . . . .  | 31        |
| 1.6.5    | Operators and vectors: a low dimensional example . . . . .                       | 33        |
| <b>2</b> | <b>Data Assimilation: state of the art</b>                                       | <b>37</b> |
| 2.1      | Variational assimilation . . . . .   | 37        |
| 2.1.1    | Maximum likelihood approach . . . . .  | 38        |
| 2.1.2    | Bayesian approach . . . . .  | 40        |
| 2.1.3    | 3D-Var scheme . . . . .  | 41        |
| 2.2      | Sequential assimilation . . . . .  | 43        |
| 2.2.1    | Kalman Filter . . . . .  | 44        |
| 2.2.2    | Extended Kalman Filter . . . . .   | 47        |
| 2.2.3    | Ensemble Kalman Filter . . . . .   | 48        |
| 2.3      | AUS: Assimilation in the Unstable Subspace . . . . .                             | 49        |
| 2.3.1    | AUS: how it works . . . . .  | 50        |
| 2.3.2    | AUS: a simple example . . . . .  | 51        |
| 2.3.3    | Refresh procedure . . . . .  | 52        |
| 2.3.4    | Using a single bred vector for assimilation . . . . .                            | 53        |
| 2.3.5    | Adaptive observation strategy . . . . .  | 53        |
| 2.4      | BDAS: Breeding on the Data Assimilation System . . . . .                         | 53        |
| 2.4.1    | Standard breeding method . . . . .   | 53        |
| 2.4.2    | BDAS: how it works . . . . .   | 54        |
| 2.4.3    | BDAS, an example of practical implementation . . . . .                           | 55        |
| <b>3</b> | <b>Assimilation in the Lorenz 63 model: comparison among different methods</b>   | <b>59</b> |
| 3.1      | Experimental setups . . . . .  | 59        |
| 3.2      | Extended Kalman Filters . . . . .  | 60        |
| 3.2.1    | EKF . . . . .  | 60        |
| 3.2.2    | Evensen's version of EKF . . . . .   | 61        |

|          |  |            |
|----------|--|------------|
| 3.2.3    | Yang's version of EKF . . . . .  | 61         |
| 3.3      | Assimilation in the Unstable Subspace: further developments . . . . .                              | 62         |
| 3.3.1    | AUS- $\gamma_0$ : no use of observations in the estimate of the forecast error amplitude . . . . . | 63         |
| 3.3.2    | Estimate of the amplitude $\gamma$ of the forecast error from observations . . . .                 | 64         |
| 3.3.3    | AUS- $\gamma$ : using the estimate of $\gamma$ from observations . . . . .                         | 65         |
| 3.3.4    | Iterating . . . . .  | 66         |
| 3.3.5    | Iterating and using a quasi-static $\mathbf{P}^f$ in stable zones of the attractor . . .           | 67         |
| 3.4      | Comparing results . . . . .  | 67         |
| 3.4.1    | Synchronization: one perfect or quasi-perfect observation, case studies . .                        | 68         |
| 3.4.2    | Noisy observations with variance $\sigma^2 = 2$ . . . . .  | 70         |
| 3.4.3    | Noisy observations with variance $\sigma^2 = 1$ . . . . .  | 81         |
| 3.4.4    | Noisy observations with variance $\sigma^2 = 0.1$ . . . . .  | 81         |
| 3.4.5    | Root Mean Square forecast error: time dependence . . . . .   | 84         |
| 3.5      | Some illustrative examples . . . . .   | 88         |
| 3.5.1    | Comparing AUS with EKF assimilation schemes: case study . . . . .                                  | 88         |
| 3.5.2    | AUS- $\gamma$ : a 3D example . . . . .   | 93         |
| 3.5.3    | AUS-iterating: a step by step description . . . . .  | 95         |
| 3.6      | Adding two types of Model Error . . . . .  | 97         |
| 3.6.1    | Random error . . . . .   | 97         |
| 3.6.2    | Random error: comparing performances . . . . .   | 98         |
| 3.6.3    | Systematic error . . . . .   | 99         |
| 3.6.4    | Systematic error: comparing performances . . . . .   | 100        |
| <b>4</b> | <b>Conclusions</b>   | <b>103</b> |
|          | <br><b>Appendices</b>  | <b>105</b> |
| <b>A</b> | <b>Euler and Runge-Kutta numerical integration methods</b>   | <b>107</b> |
| A.1      | First order Euler method . . . . .   | 107        |
| A.2      | RK2: second order Runge-Kutta scheme . . . . .   | 108        |
| A.3      | RK4: fourth order Runge-Kutta scheme . . . . .   | 109        |
| <b>B</b> | <b>Normalization factors for random variables</b>  | <b>111</b> |

|          |  |            |
|----------|--|------------|
| <b>C</b> | <b>EKF, 1-Dimensional example</b>                        | <b>113</b> |
| C.1      | Forecast and analysis covariance matrices . . . . .      | 113        |
| C.2      | Lyapunov exponents for free and forced systems . . . . . | 115        |
| <b>D</b> | <b>Inverse and pseudo-inverse matrices</b>               | <b>117</b> |
| D.1      | Inverse matrix . . . . .                                 | 117        |
| D.2      | Pseudo-inverse matrix . . . . .                          | 119        |
|          | <b>Bibliography</b>                                      | <b>121</b> |



# List of Figures

|      |   |    |
|------|---|----|
| 1.1  | A 6-hours data assimilation cycle for weather forecasts. Analyses are computed every 6 hours, typically at 0000 ZT, 0600 ZT, 1200 ZT, 1800 ZT. . . . .  | 5  |
| 1.2  | A set of Brownian processes starting at $x(0) = 0$ . The random forcing has zero mean and standard deviation $\sigma = 1$ . . . . .   | 9  |
| 1.3  | Bifurcation diagram for the Logistic map, $0 \leq r \leq 4$ . . . . .   | 17 |
| 1.4  | Bifurcation diagram for the Logistic map, $2.8 \leq r \leq 4$ . Notice the period-3 window around $r = 3.83$ . . . . .  | 18 |
| 1.5  | Bifurcation diagram for the Logistic map, a zoom. We can observe the self-similarity properties of this diagram. . . . .  | 18 |
| 1.6  | Hénon map: strange attractor for classical parameters $a$ and $b$ (see text). . . . .   | 19 |
| 1.7  | Lorenz's 1963 model: a comparison among 1-st order Euler, 2-nd and 4-th order Runge-Kutta schemes for 1000 time steps $\Delta t = 0.01$ . The initial point belongs to the attractor and is the same for all schemes: $(x_0, y_0, z_0) = (14.2041, 15.0165, 34.7172)$ . . . . . | 21 |
| 1.8  | Lorenz's 1963 model: a comparison among 1-st order Euler, 2-nd and 4-th order Runge-Kutta schemes for the variable $x$ . The initial point belongs to the attractor and is the same for all schemes: $(x_0, y_0, z_0) = (14.2041, 15.0165, 34.7172)$ . . . . .                  | 21 |
| 1.9  | Lorenz's 1963 model attractor. . . . .  | 22 |
| 1.10 | Lorenz's 1963 model attractor projected on $xz$ plane. . . . .  | 23 |
| 1.11 | Lorenz's 1963 model attractor projected on $xy$ plane. . . . .  | 23 |
| 1.12 | Lorenz's 1963 model attractor projected on $yz$ plane. . . . .  | 24 |
| 1.13 | Lorenz's 1963 model: time dependence of $x$ . . . . .   | 24 |
| 1.14 | Lorenz's 1963 model: time dependence of $y$ . . . . .   | 25 |
| 1.15 | Lorenz's 1963 model: time dependence of $z$ . . . . .   | 25 |
| 1.16 | Lorenz's 1963 map: values of the relative maximum $z_{max}(n)$ and successive relative maximum $z_{max}(n+1)$ . See Fig. 1.15. . . . .  | 26 |
| 1.17 | A simple case: 3 grid points (e, f, g) and 2 observations (1, 2). . . . .   | 34 |

|      |   |    |
|------|---|----|
| 2.1  | How the Kalman Filter works. . . . .  | 46 |
| 3.1  | Synchronization with truth for both EKF/EKF-Evensen and AUS-iterating assimilation schemes. The only observed variable is $y$ with variance $\sigma^2 = 0$ and assimilation window $\tau = 0.25$ . . . . .  | 69 |
| 3.2  | EKF and EKF-Evensen (same plot in these conditions, see text and eq. 3.40) find it hard to synchronize with truth. The only observed variable is $y$ with variance $\sigma^2 = 0$ and assimilation window $\tau = 0.6$ . . . . .                    | 70 |
| 3.3  | AUS-iterating: synchronization with truth. The only observed variable is $y$ with variance $\sigma^2 = 0$ and assimilation window $\tau = 0.6$ . . . . .  | 71 |
| 3.4  | EKF/EKF-Evensen and AUS-iterating RMS errors. While the former fail to converge to the truth, the latter synchronizes very quickly. The only observed variable is $y$ with variance $\sigma^2 = 0$ and assimilation window $\tau = 0.6$ . . . . .   | 71 |
| 3.5  | EKF/EKF-Evensen and AUS-iterating RMS errors: a zoom of the previous Fig. 3.4. . . . .  | 72 |
| 3.6  | EKF: the only observed variable is $y$ with variance $\sigma^2 = 0.01$ and assimilation window $\tau = 0.6$ ; in this case and in these conditions the EKF fails to synchronize to the truth. Note also the bad performance around time 14. . . . . | 72 |
| 3.7  | EKF-Evensen: the only observed variable is $y$ with variance $\sigma^2 = 0.01$ and assimilation window $\tau = 0.6$ . A better performance than pure EKF. . . . .   | 73 |
| 3.8  | AUS-iterating: the only observed variable is $y$ with variance $\sigma^2 = 0.01$ and assimilation window $\tau = 0.6$ . A far better performance. . . . .   | 73 |
| 3.9  | EKF, EKF-Evensen and AUS-iterating RMS errors: respective performances. The only observed variable is $y$ with variance $\sigma^2 = 0.01$ and assimilation window $\tau = 0.6$ . . . . .  | 74 |
| 3.10 | EKF, EKF-Evensen and AUS-iterating RMS errors: a zoom of the previous Fig. 3.9. . . . .   | 74 |
| 3.11 | EKF does not synchronize with truth. The only observed variable is $y$ with variance $\sigma^2 = 0.1$ and assimilation window $\tau = 0.25$ . . . . .   | 75 |
| 3.12 | EKF-Evensen does not synchronize with truth. The only observed variable is $y$ with variance $\sigma^2 = 0.1$ and assimilation window $\tau = 0.25$ . . . . .   | 75 |
| 3.13 | AUS-iterating does not synchronize with truth. The only observed variable is $y$ with variance $\sigma^2 = 0.1$ and assimilation window $\tau = 0.25$ . . . . .   | 76 |

|      |  |    |
|------|--|----|
| 3.14 | The only observed variable is $y$ with variance $\sigma^2 = 0.1$ and assimilation window $\tau = 0.25$ : in these conditions, no assimilation scheme under investigation actually converge to the truth, but AUS-iterating has a better performance than EKF-Evensen, which in turn is far better than pure EKF. . . . . | 76 |
| 3.15 | A zoom of the previous Fig. 3.14. Even if the global performance of EKF is poor because of filter divergence, this does <i>not</i> mean that EKF is <i>always</i> worse than others DA schemes for <i>all</i> times; while the global performance of AUS-iterating is globally better. . . . .                           | 77 |
| 3.16 | RMS Analysis Error distribution: an average on 100,000 assimilations with an assimilation window $\tau = 0.25$ , 3 noisy observations with variance $\sigma^2 = 2$ . . . . .   | 79 |
| 3.17 | RMS Analysis Error distribution: an average on 100,000 assimilations with an assimilation window $\tau = 0.25$ , 2 noisy observations with variance $\sigma^2 = 2$ . . . . .   | 79 |
| 3.18 | RMS Forecast Error distribution @ time= $T+0.25$ : an average on 100,000 assimilations with an assimilation window $\tau = 0.25$ , 3 noisy observations with variance $\sigma^2 = 2$ . . . . .   | 80 |
| 3.19 | RMS Forecast Error distribution @ time= $T+0.50$ : an average on 100,000 assimilations with an assimilation window $\tau = 0.25$ , 3 noisy observations with variance $\sigma^2 = 2$ . . . . .   | 80 |
| 3.20 | RMS Forecast Error distribution @ time= $T+0.75$ : an average on 100,000 assimilations with an assimilation window $\tau = 0.25$ , 3 noisy observations with variance $\sigma^2 = 2$ . . . . .   | 81 |
| 3.21 | RMS Analysis Error distribution: an average on 100,000 assimilations with an assimilation window $\tau = 0.25$ , 3 noisy observations with variance $\sigma^2 = 1$ . . . . .   | 82 |
| 3.22 | RMS Forecast Error distribution @ time= $T+0.25$ : an average on 100,000 assimilations with an assimilation window $\tau = 0.25$ , 3 noisy observations with variance $\sigma^2 = 1$ . . . . .   | 83 |
| 3.23 | RMS Forecast Error distribution @ time= $T+0.50$ : an average on 100,000 assimilations with an assimilation window $\tau = 0.25$ , 3 noisy observations with variance $\sigma^2 = 1$ . . . . .   | 83 |
| 3.24 | RMS Forecast Error distribution @ time= $T+0.75$ : an average on 100,000 assimilations with an assimilation window $\tau = 0.25$ , 3 noisy observations with variance $\sigma^2 = 1$ . . . . .   | 84 |
| 3.25 | RMS Analysis Error distribution: an average on 100,000 assimilations with an assimilation window $\tau = 0.25$ , 3 noisy observations with variance $\sigma^2 = 0.1$ . . . . .   | 85 |

|      |   |    |
|------|---|----|
| 3.26 | RMS Forecast Error distribution @ time= $T+0.25$ : an average on 100,000 assimilations with an assimilation window $\tau = 0.25$ , 3 noisy observations with variance $\sigma^2 = 0.1$ . . . . .      | 86 |
| 3.27 | RMS Forecast Error distribution @ time= $T+0.50$ : an average on 100,000 assimilations with an assimilation window $\tau = 0.25$ , 3 noisy observations with variance $\sigma^2 = 0.1$ . . . . .      | 86 |
| 3.28 | RMS Forecast Error distribution @ time= $T+0.75$ : an average on 100,000 assimilations with an assimilation window $\tau = 0.25$ , 3 noisy observations with variance $\sigma^2 = 0.1$ . . . . .      | 87 |
| 3.29 | RMS analysis and forecast error: an average on 100,000 assimilation steps, assimilation window $\tau = 0.25$ , 3 variables observed with variance $\sigma^2 = 2 \Rightarrow \sigma = 1.414$ . . . . . | 87 |
| 3.30 | RMS analysis and forecast error: an average on 100,000 assimilation steps, assimilation window $\tau = 0.25$ , 3 variables observed with variance 1. . . . .  | 88 |
| 3.31 | RMS analysis + forecast error: an average on 100,000 assimilation steps, assimilation window $\tau = 0.25$ , 3 variables observed with variance $\sigma^2 = 0.1 \Rightarrow \sigma = 0.316$ . . . . . | 89 |
| 3.32 | EKF assimilation scheme: solution for $x$ . The green line shows the time of the last observation available. . . . .  | 89 |
| 3.33 | EKF-Evensen assimilation scheme: solution for $x$ . . . . .   | 90 |
| 3.34 | EKF-Yang assimilation scheme: solution for $x$ . . . . .  | 90 |
| 3.35 | AUS- $\gamma_0$ assimilation scheme: solution for $x$ . . . . .   | 91 |
| 3.36 | AUS- $\gamma$ assimilation scheme: solution for $x$ . . . . .   | 91 |
| 3.37 | AUS-iterating assimilation scheme: solution for $x$ . . . . .   | 92 |
| 3.38 | AUS-iterating+ assimilation scheme: solution for $x$ . . . . .  | 92 |
| 3.39 | How AUS- $\gamma$ assimilates observations. . . . .   | 93 |
| 3.40 | A qualitative comparison between EKF and AUS- $\gamma$ . . . . .  | 94 |
| 3.41 | The zone of the attractor considered in the next Figure 3.42. . . . .   | 95 |
| 3.42 | How AUS-iterating works: a zoom of previous Fig. 3.41. . . . .  | 96 |
| 3.43 | More details on AUS-iterating assimilation scheme, including the evolving unstable vectors in the final forecast trajectory, recomputed after iteration. . . . .                                      | 96 |

# List of Tables

|     |  |     |
|-----|--|-----|
| 2.1 | Breeding on the Data Assimilation System: introducing the perturbations and estimating the unstable subspace. This is a specific example in which the breeding time is $\Delta t = 2\tau$ , where $\tau = t_{k+1} - t_k$ is the assimilation window. . . . .                               | 56  |
| 3.1 | The different data assimilation schemes under investigation with their main features. . . . .  | 68  |
| 3.2 | RMS analysis error, an average over 100,000 assimilations. 3 and 2 noisy observations with variance $\sigma^2 = 2 \Rightarrow \sigma = 1.414$ . Assimilation window $\tau = 0.25$ . . . .  | 78  |
| 3.3 | RMS forecast error, an average over 100,000 assimilations. The mean RMS analysis error is the same as in Table 3.2 and is shown here again for comparison. 3 noisy observations with variance $\sigma^2 = 2 \Rightarrow \sigma = 1.414$ . Assimilation window $\tau = 0.25$ . . . . .      | 78  |
| 3.4 | RMS analysis and forecast error, an average over 100,000 assimilations. 3 noisy observations with variance $\sigma^2 = 1$ . Assimilation window $\tau = 0.25$ . . . . .  | 82  |
| 3.5 | RMS analysis and forecast error, an average over 100,000 assimilations. 3 noisy observations with variance $\sigma^2 = 0.1 \Rightarrow \sigma = 0.316$ . Assimilation window $\tau = 0.25$ . . . . .   | 85  |
| 3.6 | Random error in the assimilation model: 3 observations with variance $\sigma^2 = 2 \Rightarrow \sigma = 1.414$ each assimilation window $\tau = 0.25$ . It is shown the mean RMS analysis error, an average over 20000 assimilations, for the different DA schemes. . . . .                | 98  |
| 3.7 | Random error in the assimilation model: 2 observations with variance $\sigma^2 = 2 \Rightarrow \sigma = 1.414$ each assimilation window $\tau = 0.25$ . We show the mean RMS analysis error, an average over 20000 assimilations. . . . .  | 99  |
| 3.8 | Systematic error in the assimilation model, $r = 28$ (i.e. no error, for reference), $r = 30$ , $r = 33$ . Three observations with variance $\sigma^2 = 2$ each assimilation window $\tau = 0.25$ . It is shown the mean RMS analysis error, an average over 20,000 assimilations. . . . . | 100 |

|     |  |     |
|-----|--|-----|
| 3.9 | Systematic error in the assimilation model, $r = 28$ (i.e. no error, for reference),<br>$r = 30$ , $r = 33$ . Two observations with variance $\sigma^2 = 2$ each assimilation<br>window $\tau = 0.25$ . It is shown the mean RMS analysis error, an average over<br>20000 assimilations. . . . . | 101 |
|-----|--|-----|

# Abbreviations

|        |  |
|--------|--|
| 3D-Var | 3-dimensional variational analysis                 |
| 4D-Var | 4-dimensional variational analysis                 |
| AUS    | Assimilation in the Unstable Subspace              |
| BDAS   | Breeding on the Data Assimilation System           |
| BV     | Bred Vector  |
| CFL    | Courant-Friedricks-Lewy                            |
| DA     | Data Assimilation                                  |
| ECMWF  | European Centre for Medium-Range Weather Forecasts |
| EKF    | Extended Kalman Filter                             |
| EnKF   | Ensemble Kalman Filter                             |
| ENIAC  | Electronic Numerical Integrator And Computer       |
| FPE    | Fokker-Planck Equation                             |
| GCM    | General Circulation Model                          |
| KF     | Kalman Filter                                      |
| L63    | Lorenz's 1963 three variables convective systems   |
| LAM    | Local Area Model                                   |
| LE     | Liouville Equation                                 |
| LLV    | Local Lyapunov vector                              |
| LT     | Liouville Theorem                                  |

|     |   |
|-----|---|
| LV  | Lyapunov vector                                     |
| NWP | Numerical Weather Prediction                        |
| OI  | Optimal Interpolation                               |
| PDF | Probability Density Function                        |
| QG  | Quasi-Geostrophic                                   |
| RK  | Runge-Kutta   |
| RK2 | 2nd order Runge-Kutta scheme                        |
| RK4 | 4-th order Runge-Kutta scheme                       |
| RMS | Root Mean Square                                    |
| SCM | Successive Correction Method                        |
| TLM | Tangent Linear Model                                |
| UTC | Universal Time, Coordinated. Also referred to as ZT |
| ZT  | Zulu Time. Also referred to as UTC                  |



# Partial list of symbols

Symbols are listed in alphabetical order, first Latin alphabet, then Greek one. Uppercase symbols precede lowercase ones.

|                |  |
|----------------|--|
| <b>E</b>       | matrix whose columns are the unstable vectors                |
| $H$            | observation operator   |
| <b>H</b>       | linearized observation operator                              |
| <b>I</b>       | identity matrix  |
| <b>K</b>       | gain matrix  |
| $M$            | nonlinear model operator                                     |
| <b>M</b>       | linear model operator, tangent linear model operator         |
| $\mathbf{P}^a$ | analysis error covariance matrix                             |
| $\mathbf{P}^f$ | forecast error covariance matrix                             |
| <b>Q</b>       | model error covariance matrix                                |
| <b>R</b>       | observational error covariance matrix                        |
| $S^{Mod}$      | model space  |
| $S^{ob}$       | observational space  |
| <b>e</b>       | leading local Lyapunov vector, single unstable column vector |
| $f$            | generic function, Coriolis parameter                         |
| $\mathbf{x}^a$ | analysis state column vector                                 |
| $\mathbf{x}^b$ | background, or “first guess” state column vector             |
| $\mathbf{x}^f$ | forecast state column vector                                 |

|                       |  |
|-----------------------|--|
| $\mathbf{y}^b$        | background, or “first guess” values at observation points  |
| $\mathbf{y}^o$        | observation vector (includes noise) column vector  |
| $\mathbf{y}^t$        | true values of the observed variables, column vector   |
| $\mathbf{\Gamma}$     | diagonal matrix whose elements are the estimate of the amplitudes of the errors along the unstable vectors |
| $\mathbf{\Lambda}$    | diagonal matrix whose elements are the amplification factors of the unstable vectors                       |
| $\mathbf{\Omega}$     | earth angular rotation vector  |
| $\delta\mathbf{x}^a$  | analysis increment $\mathbf{x}^a - \mathbf{x}^b$   |
| $\delta\mathbf{y}^o$  | innovation $\mathbf{y}^o - H(\mathbf{x}^b)$  |
| $\gamma$              | estimate of error amplitude  |
| $\boldsymbol{\eta}^M$ | Model error  |
| $\boldsymbol{\eta}^a$ | analysis error $\mathbf{x}^a - \mathbf{x}^t$   |
| $\boldsymbol{\eta}^f$ | forecast error $\mathbf{x}^f - \mathbf{x}^t$   |
| $\boldsymbol{\eta}^o$ | observational error $\mathbf{y}^o - H(\mathbf{x}^t)$   |
| $\lambda_i$           | $i$ -th Lyapunov exponent  |
| $\tau$                | assimilation window  |

# Chapter 1

## Introduction

The Atmosphere plays an important role in weather and climate systems, including the hydrosphere (Oceans), cryosphere (ice and snow), lithosphere (soil) and biosphere (living systems). All these components interact with each others [8]. The Atmosphere is also well known to be a chaotic system with an enormous number of degrees of freedom and with many scales of motions, both in space and time: its predictability has a finite time horizon [31]. In this context, to accurately estimate the state of the system we need to extract as much information as possible from observational data, when available, and from the equations governing the evolution of the system, to the extent they are known: that's the goal of *Data Assimilation*, a fundamental step for Numerical Weather Prediction (NWP). In order to develop new theoretical work, we address the problem of data assimilation in chaotic systems in the framework of Lorenz's three variable convective model (1963). The aim is to develop data assimilation schemes applicable to different contexts and computationally affordable in operational environments.

### 1.1 A historical perspective

It was a Norwegian meteorologist, Vilhelm Bjerknes, who first realized in 1904 that weather forecasts were actually an initial value problem, to be solved by integrating the set of governing equations of the atmosphere, starting from an accurate estimate of initial conditions obtained from observations. This was the first explicit recognition that the future state of the atmosphere can be completely, deterministically calculated by its initial state and known boundary conditions, together with seven equations: Newton's equations of motion (three equations for the three velocity components), the continuity equation (conservation of mass), the equation of state for ideal gases, the first law of thermodynamics (conservation of energy) and a conservation equation for water mass [16]. Bjerknes was able to persuade Norwegians to build

a network of surface observation stations, founded the renown Bergen School of synoptic and dynamic meteorology and proposed the famous polar front theory of cyclogenesis.

Lewis F. Richardson suggested in 1922 to integrate numerically the equations of motion of the atmosphere and described how exactly this could be done. His first attempt of weather forecast was actually unsuccessful, but did not diminish the value of his seminal work. The increasing development of observational network on one hand, and the introduction of reliable computing machines on the other, boosted a new interest on Richardson’s approach.

An important issue to address was balancing of initial conditions: if they are not in quasi-geostrophic balance<sup>1</sup>, inertia-gravity waves will arise and propagate horizontally. After a while, they will drastically reduce their amplitude, leaving a field in quasi-geostrophic balance: this process is called *geostrophic adjustment*. The time scale for this process to take place is of the order<sup>2</sup> of  $f^{-1}$ , approximately 12h [16]. Moreover, in Richardson’s first attempt, the integration of the equations resulted in computational instability, due to a violation of Courant-Friedricks-Lewy (CFL) condition, which requires that the time step must be smaller than the grid size divided by the speed of the fastest waves (sound waves, moving at about 340 m/s).

To address these problems, in 1948-49 Jule C. Charney and Eliassen introduced “filtered” equations of motion, based on quasi-geostrophic balance — i.e. slowly varying, so filtering out gravity and sound waves — and based on pressure fields alone. In 1950 Charney, R. Fjørtoft and J. von Neuman performed on ENIAC, one of the first computers built<sup>3</sup>, the first historical 24h weather forecast using a barotropic one-layer filtered model: the results were encouraging. The initial values, though, were still set by *subjective analysis*, relying on judgments of experienced analysts. In 1949 Panofsky made the first attempt to overcome subjective analysis by an automatic procedure, called *objective analysis*, albeit the term “objective” actually depends on the algorithms used. This procedure used a polynomial expansion to fit all the observations of several grid-points in a given area.

The first operational numerical weather forecast were issued in Sweden by Rossby and his group in September 1954. In the meanwhile, new advances were introduced by Gilchrist and Cressman (1954), who used a polynomial expansion, too, but with a local rather than areal

---

<sup>1</sup>The quasi-geostrophic equations hold for large scale, low-frequency motions, except low latitudes [1].

<sup>2</sup>The Coriolis parameter  $f \approx 2\Omega \sin \varphi_0$ , where  $\Omega$  is the earth’s angular velocity, and  $\varphi_0$  the latitude.

<sup>3</sup>The ENIAC (Electronic Numerical Integrator And Computer) was built in 1946. It was indeed not the first computer ever built and operating. The first fully functional, freely programmable computers of the world was actually the German Konrad Zuse’s Z1, Z2 and Z3, electro-mechanical machines built between 1936 and 1941. The next generation used vacuum tubes: the Atanasoff-Berry Computer, built in 1939 at Iowa State University, USA, and the British *Colossus*, operating since December 1943 in Bletchley Park, Milton Keynes, UK. Both Zuse and Atanasoff-Berry computers were conceptual milestones in Computer Science, because the former introduced the idea of programmable machines and the latter the use of the binary system and other innovations. Nevertheless, it was the *Colossus* that had a truly enormous impact in the mankind history, because it helped a lot in defeating Nazism: it was used during World War II to decrypt the most important messages transmitted between the German Army Field marshals and their Central High Command in Berlin.

fit of observations: the idea of *radius of influence* was born. At each grid point they used a polynomial function to approximate the fields, taking into account only the observations near the grid point, i.e. within the radius of influence. Two more elements, introduced by the same authors, were later adopted in successive works: automatic check for data quality, and an *a priori* estimate of the analysis, obtained from a previous numerical forecast. This preliminary estimate is now referred to as *background field*, *first guess field*, or *prior estimate*.

During these pioneering efforts by meteorologists to run reliable NWP models, it quickly turned out that the accuracy of a model strongly depends on spatial resolution. In general, the higher the resolution, the higher the accuracy of the model, but — of course — the higher the computational cost as well. That’s because, due to computational stability requirements, doubling the 3-dimensional space resolution also requires to double the time resolution. This implies a  $2^4$  factor for the total computational cost (three spatial and one time dimensions). As a result, grinding atmospheric models has always been a challenging task for supercomputers, and the model size has always been driven, in turn, by the available computing capacity.

Further improvements came along: in 1955 Bergthorsson and Doos developed an analysis method that eventually became known as “*successive correction*”. They reduced the computational cost of the interpolating procedures by specifying an *a priori* weight for each observation, weight to be determined on a statistical basis.

Thompson in 1961 proposed to take full advantage of the propagation of information from well observed regions to data-void ones. A quite usual situation in the global observing system is the presence of zones where new observations are regularly available (e.g. densely populated regions) and regions where new observational data are scarce (e.g. oceans and deserts). An objective analysis can be made for the former areas, and some sort of educated guess for the latter. Then the integration of a NWP model will provide a forecast valid at the next observation time. Now we have new observations from data-rich areas and model-output data for data-poor ones: information has propagated.

When computers became sufficiently fast, scientists turned to primitive equations instead of filtered ones, and introduced regional models (or Local Area Models, LAMs) side by side to General Circulation Models (GCMs), a GCM setting the boundary conditions for the regional one. Thus, a regional model can be used for short range forecast only, as its high quality initial conditions will be lost due to the “information advection” of the ever changing boundary conditions driven by the GCM. Furthermore, as a consequence of increasing computing power, now meteorologists no more tend to use the hydrostatic approximation — in which vertical acceleration is neglected versus the gravitational one — for regional models.

## 1.2 Data Assimilation

In his pioneering paper cited in the section above [4], Charney stressed the need of objective analysis of meteorological data, not to rely upon time-consuming human activities and subjective interpretations. This process is nowadays referred to as *Data Assimilation* (DA): its goal is to produce an optimal, automatic estimate of the state of a dynamical system from incomplete, noisy observations and (approximate) knowledge of the laws governing the evolution of the system. In the initialization of forecast models of the Ocean and the Atmosphere, DA is performed cyclically. A new term, *Data Assimilation Cycle*, has been introduced to describe this cyclic procedure, that encompasses the following steps:

- Quality control of observational data
- Objective analysis
- Initialization of the forecast model
- Short-range forecast to be used to estimate the next background field

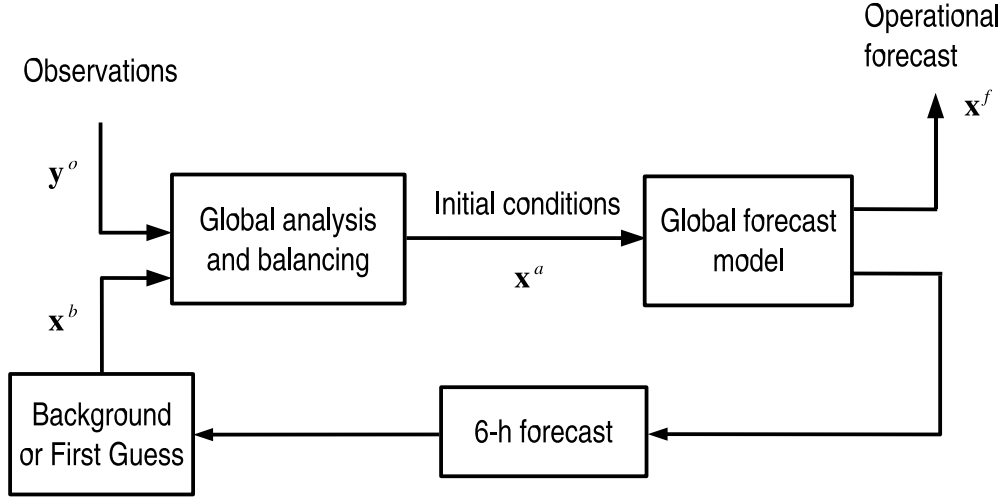
Quality control is a very delicate step, because it has been shown that the analysis can be highly sensitive to quality control decisions [8]. Generally speaking, datum quality is checked against its neighbors, and a further requirement of spatial and temporal consistency is asked to be fulfilled. Observations can be checked against the background, too.

Objective analysis step exploits both the available observational data,  $\mathbf{y}^o$ , and the background field,  $\mathbf{x}^b$ , numerically computed in the previous observation time. Obtaining the background or first guess “observations” is a matter of interpolating the model forecast to the observational stations: model variables are converted to observed variables. First guess “observations” are so  $H(\mathbf{x}^b)$ , where  $H$  is the operator performing the required interpolation and conversion from model variables to observation space. It should be noted that the operator  $H$  is not a linear one, in general. The difference  $\mathbf{y}^o - H(\mathbf{x}^b)$  between the observations and the first guess “observations” is usually called *observational increment* or *innovation* [16]. The analysis state  $\mathbf{x}^a$  is computed by adding the innovation to the model background field, with weights  $W$  to be determined by estimating statistical error covariances of the forecast and the observations:

$$\mathbf{x}^a = \mathbf{x}^b + W [\mathbf{y}^o - H(\mathbf{x}^b)] \quad (1.1)$$

Many analysis techniques, such as Successive Correction Method (SCM), Optimal Interpolation (OI), 3-dimensional variational analysis (3D-Var) and Kalman Filter (KF) use eq. 1.1, but with different ways to calculate the weights  $W$ .

Figure 1.1: A 6-hours data assimilation cycle for weather forecasts. Analyses are computed every 6 hours, typically at 0000 ZT, 0600 ZT, 1200 ZT, 1800 ZT.



After calculating the analysis state  $\mathbf{x}^a$ , the *forecast model* can be initialized to obtain the routine forecast  $\mathbf{x}^f$ : this is actually the aim of the entire procedure. The numerical short-range forecast to estimate the background field for the next observation time is usually the output of a high resolution model that implements primitive equations. This model, called *assimilation model*, has a complex set of parameterizations such that, if no new observations become available, the model climate — computed by time-averaging a long run of the model — will approximate the true climate. Short-range numerical forecast, typically 6-h ahead, replaced the simple use of climatology as background field.

One approach to assimilate observations at various time is 4D-Var (e.g. Lewis and Derber [17], Courtier and Talagrand [7]). This assimilation system is the operational DA scheme used at ECMWF and Météo-France, among other important meteorological centers.

### 1.3 Framing the problem

This work is focused in data assimilation on chaotic systems. In order to introduce chaotic systems, in this section we will briefly describe the general properties of conservative and nonconservative systems; then we will survey the basics of dissipative and chaotic systems, attractors,

Lyapunov vectors and exponents, and the bred vectors. We will also provide classic examples of chaotic dynamical systems, described both by differential equations and maps.

### 1.3.1 Basic features of chaotic systems, a historical note

The fundamental property of a chaotic system is the *sensitive dependence on initial conditions*, discovered by Jules-Henri Poincaré in 1897 for a simplified *three-body problem*: a planetary system including 2 stars and a small “asteroid” [16]. Later in a remarkable 1908 monograph — *Science et Méthode* — he wrote down these basic concepts about chaos (Livre premier, § IV):

*“A very small cause which escapes our notice determines a considerable effect that we cannot fail to see, and then we say that the effect is due to chance. If we knew exactly the laws of nature and the situation of the universe at the initial moment, we could predict exactly the situation of the same universe at a succeeding moment. But even if it were the case that the natural laws had no longer any secret for us, we could still know the situation only **approximately**. If that enables us to predict the succeeding situation **with the same approximation**, that is all we require, and we will say that the phenomenon has been predicted, that it is governed by the laws. But it is not always so; it may happen that small differences in the initial conditions produce very great ones in the final phenomena. A small error in the former will produce an enormous error in the latter. Prediction becomes impossible and we have the fortuitous phenomenon.”* – Jules-Henri Poincaré, 1908

Albeit this note was written in 1908, it has never become dated. Thanks to the increasing computational capability of computers, the meteorologist Edward N. Lorenz, now Professor Emeritus at Massachusetts Institute of Technology, “rediscovered” chaos in early 60s while examining a relatively simple mathematical model of weather. That’s the same we will survey in section 1.4 below.

### 1.3.2 Conservative, nonconservative dynamical systems

Generally speaking, a *dynamical system* changes its state depending on time. A dynamical system that evolves continuously in time is known as a *flow* [19]. It can be described by a set of differential equations giving the evolution of the state of the system, knowing its previous states:

$$\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}) \tag{1.2}$$



where  $\mathbf{x}(t) = (x_1(t), x_2(t), \dots, x_N(t))$  is the  $N$ -dimensional vector — depending on time  $t$  — describing the state of the system at time  $t$ : the vector  $\mathbf{x}(t)$  can be unambiguously mapped, through a bijection, to a point in the phase space of the system. The term  $\dot{\mathbf{x}} = \frac{d\mathbf{x}}{dt}$  is the time derivative of  $\mathbf{x}(t)$  and  $\mathbf{F}(\mathbf{x}) = (F_1(\mathbf{x}), F_2(\mathbf{x}), \dots, F_N(\mathbf{x}))$  is a  $N$ -dimensional vector function of the state  $\mathbf{x}$  of the system. Given the initial state  $\mathbf{x}(0)$  we can deterministically calculate the *trajectory*, or *orbit*,  $\mathbf{x}(t)$  of the system for all future times. Here the time variable  $t$  is continuous.

In a system where particles move without friction, called Hamiltonian system, the Liouville's theorem (see subsection 1.3.5) assures that the volume of any subset of points in the phase space is conserved: if each point of the initial subset is evolved forward in time, the resulting set of points has the same volume as the initial one. So the system is also called *conservative*. In a *nonconservative* system, instead, time evolution does not preserve volumes in phase space.

### 1.3.3 Probability Density Functions and Liouville equation

In a dynamical system of the form  $\frac{d\mathbf{x}}{dt} = \mathbf{F}(\mathbf{x}, t)$ , there could be uncertainties due to a bad estimation of initial conditions, or even a bad knowledge of the model used: a statistical approach is a standard practice. The *probability density function* (PDF) associated to the continuous variable  $\mathbf{x}(t)$  can be thought of as the set of possible realizations in the phase space of the outcome of the model:

$$\rho = \rho(\mathbf{x}(t), t) \quad (1.3)$$

If the PDF is Gaussian — an usual assumption in data assimilation — it is completely defined by the mean of the state  $\mathbf{x}(t)$  and by the second moment about the mean, i.e. its variance:

$$\langle \mathbf{x} \rangle \equiv \int \dots \int_{-\infty}^{+\infty} \mathbf{x} \rho(\mathbf{x}(t), t) d\mathbf{x} \quad (1.4)$$

$$\langle (\mathbf{x} - \langle \mathbf{x} \rangle)^2 \rangle \equiv \int \dots \int_{-\infty}^{+\infty} (\mathbf{x} - \langle \mathbf{x} \rangle)^2 \rho(\mathbf{x}(t), t) d\mathbf{x} \quad (1.5)$$

The *Liouville equation* (LE) is the probabilistic description of the time-dependent evolution of an ensemble of solutions of the numerical model  $\frac{d\mathbf{x}}{dt} = \mathbf{F}(\mathbf{x}, t)$  from different initial conditions [9]. It governs the time evolution of the PDF  $\rho(\mathbf{x}(t), t)$  associated to the model state  $\mathbf{x}(t)$ . It may be written in a simplified form as follows:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{F} = 0 \quad (1.6)$$

or in the more complete form:

$$\frac{\partial \rho(\mathbf{x}(t), t)}{\partial t} + \sum_{i=1}^N \frac{\partial}{\partial x^i} [\rho(\mathbf{x}(t), t) F^i(\mathbf{x}(t), t)] = 0 \quad (1.7)$$

where  $F^i$  is the  $i$ -th component of the vector function  $\mathbf{F}$ . The Liouville equation is a conservation equation: it states that the local change of  $\rho$  — in a particular point of the phase space — must be equal to the net flux of realizations across the faces of an infinitesimal volume around the point under examination; equivalently, Liouville equation means that phase space integral of the realization density is a constant with respect to time. It is an inhomogeneous partial differential equation, linear in the PDF  $\rho$ , which is the single dependent variable.

### 1.3.4 The Markov processes and the Fokker-Planck equation

If our dynamical system is forced by some sort of stochastic noise, that could arise from our model's misrepresentation, the general equation reads:

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{F}(\mathbf{x}(t), \mathbf{q}(t), t) \quad (1.8)$$

where  $\mathbf{q}(t)$  is the vector of random disturbances. If  $\mathbf{q}(t)$  is a Markov process its probability law in the future does only depend on the given state, not on how the system reached that state. An example of Markov process is the Brownian motion

$$x(t_n) = x(t_{n-1}) + w(t_n) \quad (1.9)$$

where  $w$  is a white Gaussian forcing. A few examples of Brownian motions, with standard deviation of the Gaussian forcing  $\sigma = 1$ , are shown in Fig. 1.2.

Furthermore, if in eq. 1.8  $\mathbf{q}(t)$  represents an additive white Gaussian forcing function, we can write [15]:

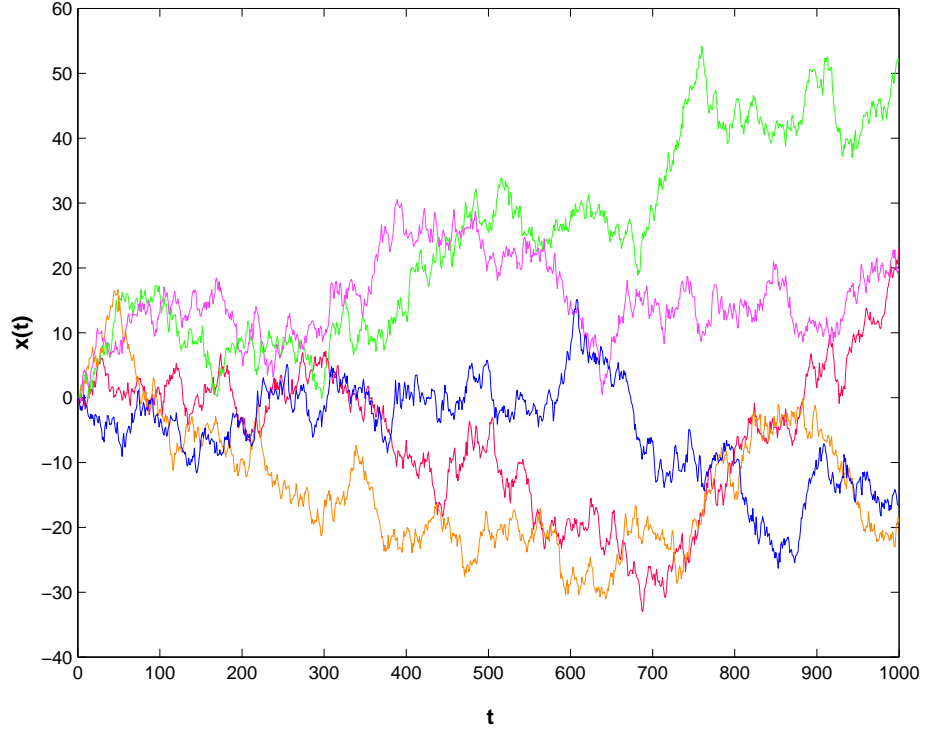
$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{F}(\mathbf{x}(t), t) + \frac{d\mathbf{q}(t)}{dt} \quad (1.10)$$

because the white Gaussian noise can be thought of as the derivative of Brownian motion [15]. Equation 1.10 is sometimes called the *Langevin equation*. It can be shown that, being  $\mathbf{q}(t)$  a Markov process, so it is  $\mathbf{x}(t)$ . We can now write the solution of 1.10 in the form

$$\mathbf{x}(t) = M[\mathbf{x}(t_0)] + \mathbf{q}(t) \quad (1.11)$$

The *Fokker-Planck equation* (FPE) describes the evolution of the PDF associated to these

Figure 1.2: A set of Brownian processes starting at  $x(0) = 0$ . The random forcing has zero mean and standard deviation  $\sigma = 1$ .



stochastic systems: it includes a random term — such as for example a model error — in the Liouville equation 1.7. This term has the form of a diffusion component:

$$\frac{\partial \rho(\mathbf{x}(t), t)}{\partial t} = - \sum_{i=1}^N \frac{\partial [\rho(\mathbf{x}(t), t) F^i(\mathbf{x}(t))]}{\partial x^i} + \frac{1}{2} \sum_{i,j=1}^N \frac{\partial^2 [\rho(\mathbf{x}(t), t) (Q)^{i,j}]}{\partial x^i \partial x^j} \quad (1.12)$$

where the first sum refers to the drift and the second to diffusion: the Fokker-Planck equation describes the time evolution of a PDF due to both of them. The matrix  $\mathbf{Q}$  is the stochastic noise covariance matrix. It should be noted that the FPE is a linear in  $\rho$ , which is the only dependent variable.

### 1.3.5 The Liouville Theorem

As we already mentioned in subsection 1.3.2, the *Liouville Theorem* (LT) states that a conservative system  $\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x})$  conserves in time the volumes of any subset of points in the phase space. In nonconservative systems, instead, this does not occur: in particular, as we will prove below, an initial volume  $V_0 = V(0)$  of a given phase space region  $D_0$  shrinks according to [26, 2]:

$$\left. \frac{dV}{dt} \right|_{t=0} = \int_{D_0} \nabla \cdot \mathbf{F}(\mathbf{x}) d\mathbf{x} \quad (1.13)$$

If  $\nabla \cdot \mathbf{F}(\mathbf{x})$  does not depend on the vector  $\mathbf{x}$ , eq. 1.13 simplifies:

$$\left. \frac{dV}{dt} \right|_{t=0} = \nabla \cdot \mathbf{F} \int_{D_0} d\mathbf{x} \quad (1.14)$$

$$\frac{dV_0}{dt} = V_0 \nabla \cdot \mathbf{F} \quad (1.15)$$

Rearranging and integrating from time  $t_0 = 0$  to  $t$ :

$$\int_{V_0}^V \frac{dV_0}{V_0} = \nabla \cdot \mathbf{F} \int_{t_0}^t dt \quad (1.16)$$

$$\ln \frac{V}{V_0} = t \nabla \cdot \mathbf{F} \quad (1.17)$$

So finally, in this particular case:

$$V(t) = V_0 e^{t \nabla \cdot \mathbf{F}} \quad (1.18)$$

**Proof** of eq. 1.13. In the phase space of a dynamical system described by equation 1.2 we can define the *phase flux*  $g^t$ :

$$g^t : \mathbf{x}(0) \rightarrow \mathbf{x}(t) \quad (1.19)$$

and, by definition of Jacobian  $\frac{\partial g^t \mathbf{x}}{\partial \mathbf{x}}, \forall t$  we have:

$$V(t) = \int_{D_0} \det \left( \frac{\partial g^t \mathbf{x}}{\partial \mathbf{x}} \right) d\mathbf{x} \quad (1.20)$$

For  $t \rightarrow 0$ :

$$g^t(\mathbf{x}) = \mathbf{x} + \mathbf{F}(\mathbf{x}) t + \mathcal{O}(t^2) \quad (1.21)$$

Thus

$$\frac{\partial g^t \mathbf{x}}{\partial \mathbf{x}} = \mathbf{I} + \frac{\partial \mathbf{F}}{\partial \mathbf{x}} t + \mathcal{O}(t^2) \quad (1.22)$$

But for any  $N \times N$  square matrix  $\mathbf{A} = (a_{ij})$  and for  $t \rightarrow 0$  it holds the following relation:

$$\det(\mathbf{I} + \mathbf{A}t) = 1 + t \operatorname{tr}(\mathbf{A}) + \mathcal{O}(t^2) \quad (1.23)$$

where  $\operatorname{tr}(\mathbf{A}) = \sum_{i=1}^N a_{ii}$  is the *trace* of the matrix  $\mathbf{A}$ . So we have:

$$\det \left( \frac{\partial g^t \mathbf{x}}{\partial \mathbf{x}} \right) = 1 + t \operatorname{tr} \left( \frac{\partial \mathbf{F}}{\partial \mathbf{x}} \right) + \mathcal{O}(t^2) \quad (1.24)$$

Now we notice that, of course:

$$\text{tr} \left( \frac{\partial \mathbf{F}}{\partial \mathbf{x}} \right) = \sum_{i=1}^N \frac{\partial F_i}{\partial x_i} = \nabla \cdot \mathbf{F} \quad (1.25)$$

so eq. 1.20 becomes:

$$V(t) = \int_{D_0} [1 + t \nabla \cdot \mathbf{F} + \mathcal{O}(t^2)] d\mathbf{x} \quad (1.26)$$

which proves equation 1.13. If the divergence  $\nabla \cdot \mathbf{F} = 0$ , the phase flux  $g^t$  preserves the volumes:  $\forall t, V(t) = V(0)$  and the Liouville theorem has been proved. Because of that volume independence on time, the system is called *conservative*.

### 1.3.6 Dissipative systems, attractors and strange attractors

Systems which exhibit volume contraction in phase space are called *dissipative*, because commonly friction, viscosity or other processes dissipating energy are involved. The volume contraction proves also the existence of a bounded attracting set of points, the *attractor*, toward which converge all trajectories, after an appropriate transient time: if we consider initial conditions in an adequate region of phase space, for increasing time  $t$  they will eventually converge to the attractor.

More formally, an attractor is a closed set  $A$  satisfying the following properties [31]:

- $A$  is an *invariant set*: any trajectory  $\mathbf{x}(t)$  starting in  $A$  will remain in  $A$  for all time:  
 $\forall \mathbf{x}(t), \mathbf{x}(0) \in A \Rightarrow \mathbf{x}(t) \in A, \forall t$
- $A$  *attracts an open set of initial conditions*: this means that  $A$  attracts all trajectories starting sufficiently close to it:  $\exists B$ , with  $B$  an open set, so that  $\mathbf{x}(0) \in B \Rightarrow d(\mathbf{x}(t), A) \rightarrow 0$  as  $t \rightarrow \infty$ , where  $d(\mathbf{x}(t), A)$  is the distance from  $\mathbf{x}(t)$  to  $A$ . The largest  $B$  is called the *basin of attraction* of  $A$
- $A$  is *minimal*: no proper subset of  $A$  will satisfy the above properties

In many cases, as for example the Lorenz's 1963 convective system (see below), we have a *strange attractor*: the “strangeness” refers to the *sensitive dependence on initial conditions* of the nonperiodic flow, though initially strange attractors were called in such a way because of their common fractal dimensionality [31]. A solution which is *stable in the sense of Lyapunov* means that any other solution sufficiently close to it will remain close for increasing time. Thus “sensitive dependence on initial conditions” means actually “unstable in the sense of Lyapunov”. It can be shown that a solution possessing Lyapunov stability must be a periodic or quasi-periodic one [19].

The Lorenz's 1963 system clearly shows a lack of periodicity, as we can see for example in Fig. 1.13 below, which in turn implies a limited predictability of the system because of sensitive dependence on initial conditions [19]. The general behavior is chaotic, even if we can always find unstable periodic orbits arbitrarily close to aperiodic ones [32].

### 1.3.7 Small perturbations dynamics, tangent linear model, adjoint model

Lyapunov instability is a matter of small perturbation growth. A small perturbation  $\delta\mathbf{x}(t)$  of a trajectory  $\mathbf{x}(t)$  is assumed to evolve in a linear way. That is:

$$\delta\mathbf{x}(t_{k+1}) = \mathbf{M}_k \delta\mathbf{x}(t_k) \quad (1.27)$$

which is the *Tangent Linear Model* (TLM). Here the operator  $\mathbf{M}_k$ , that depends on time  $t_k$ , is an operator linearized around the base-trajectory. It is called the *resolvent*, or *propagator*, of the TLM. Since  $\mathbf{M}_k$  is an operator defined on real numbers, its *adjoint* is simply its transpose, the operator  $\mathbf{M}_k^T$ .

In order to justify eq. 1.27, consider a nonlinear discrete model that can be written as a set  $N$  nonlinear coupled ordinary differential equations

$$\frac{d\mathbf{x}}{dt} = \mathbf{F}(\mathbf{x}) \quad (1.28)$$

where  $\mathbf{x}$  is an  $N$ -dimensional vector and  $\mathbf{F}$  an  $N$ -dimensional vector function. The model is written in differential form. When a time-difference scheme is chosen, eq. 1.28 becomes a set of *difference* equations. If for example a Crank-Nicholson approach is implemented, this set of equations would be of the form [16]:

$$\mathbf{x}(t_{k+1}) = \mathbf{x}(t_k) + \Delta t \cdot \mathbf{F} \left( \frac{\mathbf{x}(t_k) + \mathbf{x}(t_{k+1})}{2} \right) \quad (1.29)$$

So we can integrate eq. 1.28 running the model between an initial time  $t_0$  and a final time  $t$ , by recursively using eq. 1.29: the solution  $\mathbf{x}(t)$  will depend on initial conditions only:

$$\mathbf{x}(t) = M [\mathbf{x}(t_0)] \quad (1.30)$$

which depends only on time  $t_0$ . Here the operator  $M$  — that in general is nonlinear — represents the time integration between  $t_0$  and  $t$ .

If we add a small perturbation  $\delta\mathbf{x}(t_0)$  to the reference model integration  $\mathbf{x}(t_0)$  we can write:

$$M[\mathbf{x}(t_0) + \delta\mathbf{x}(t_0)] = M[\mathbf{x}(t_0)] + \frac{\partial M}{\partial \mathbf{x}} \delta\mathbf{x}(t_0) + \mathcal{O}[\delta\mathbf{x}(t_0)^2] \quad (1.31)$$

$$= \mathbf{x}(t) + \delta\mathbf{x}(t) + \mathcal{O}[\delta\mathbf{x}(t_0)^2] \quad (1.32)$$

where we are using 1.30 and the small perturbation dynamics:

$$\delta\mathbf{x}(t) = \frac{\partial M}{\partial \mathbf{x}} \delta\mathbf{x}(t_0) \quad (1.33)$$

$$= \mathbf{M} \delta\mathbf{x}(t_0) \quad (1.34)$$

that is the same as eq. 1.27. Here  $\mathbf{M} = \frac{\partial M}{\partial \mathbf{x}}$  is the  $N \times N$  matrix called the resolvent or propagator of the tangent linear model, and propagates an initial small perturbation at time  $t_0$  to a perturbation at time  $t$ . Since it is linearized from  $t_0$  to  $t$ ,  $\mathbf{M}$  depends on reference trajectory  $\mathbf{x}(t)$  but not on perturbation  $\delta\mathbf{x}(t_0)$ . The linearized evolution of  $\delta\mathbf{x}(t_0)$  will be given by

$$\frac{d\delta\mathbf{x}(t)}{dt} = \frac{\partial \mathbf{F}[\mathbf{x}(t)]}{\partial \mathbf{x}} \delta\mathbf{x}(t) \quad \forall t \in [t_0, t] \quad (1.35)$$

where  $\frac{\partial \mathbf{F}[\mathbf{x}(t)]}{\partial \mathbf{x}}$  is the Jacobian of  $\mathbf{F}$ . This system (eq. 1.35) defines the tangent linear model in differential form [16].

### 1.3.8 Lyapunov vectors and Lyapunov exponents

In order to have a more precise, quantitative idea of the “sensitive dependence on initial conditions”, we will summarize the concepts of Lyapunov vectors and Lyapunov exponents.

Let’s consider a trajectory on the attractor (after an appropriate transient time): here a state at time  $t$  is described by the vector  $\mathbf{x}(t)$ . Now we consider a very close point,  $\mathbf{x}(t) + \delta\mathbf{x}(t)$ , where  $\delta\mathbf{x}(t)$  is a separation vector whose initial length  $\delta\mathbf{x}(0)$  is very small. We are interested in how  $\delta\mathbf{x}(t)$  will grow. One finds [31] that close trajectories, starting on a sphere of infinitesimal radius, diverge exponentially fast:

$$\|\delta\mathbf{x}(t)\| \simeq \|\delta\mathbf{x}(0)\| e^{\lambda t} \quad (1.36)$$

where  $\lambda$  is called the global *leading* (or *largest*) *Lyapunov exponent*. It describes the long term growth of the resulting hyper-ellipsoid, and can be estimated by [16]:

$$\lambda = \lim_{t \rightarrow +\infty} \frac{1}{t} \left[ \lim_{\delta\mathbf{x}(0) \rightarrow 0} \ln \left\| \frac{\delta\mathbf{x}(t)}{\delta\mathbf{x}(0)} \right\| \right] \quad (1.37)$$

In practice, the leading Lyapunov exponent is computed as follows:

- we perturb the vector trajectory  $\mathbf{x}(t)$  with an infinitesimal random vector  $\delta\mathbf{x}(t)$
- we evolve it from time  $t$  to time  $t + \Delta t$  using the Tangent Linear Model:

$$\delta\mathbf{x}(t + \Delta t) = \mathbf{M} \delta\mathbf{x}(t) \quad (1.38)$$

where we dropped — for sake of lighter notations — the obvious dependence of  $\mathbf{M}$  on  $t$  and  $\Delta t$

- we repeat the previous step for a long time, scaling down at regular intervals the perturbation vector to avoid computational overflow

Other Lyapunov exponents can be computed in the same way, except that we must periodically perform a Gramm-Schmidt orthogonalization to the set of perturbations that defines the shrinking hyper-ellipsoid: otherwise, they all will converge to the first Lyapunov exponent. It should be noted, indeed, that for an  $N$ -dimensional system there are  $N$  Lyapunov exponents: an initially infinitesimal  $N$ -dimensional hyper-sphere will be distorted, due to the evolution of the system, in an infinitesimal hyper-ellipsoid. If we define  $\delta_i\mathbf{x}(t)$ ,  $i = 1, \dots, N$  the length of the  $N$  principal axis of this hyper-ellipsoid, then  $\lambda_i$  will be their growth rates, and equation 1.36 will be replaced by:

$$\|\delta_i\mathbf{x}(t)\| \simeq \|\delta_i\mathbf{x}(0)\| e^{\lambda_i t} \quad (1.39)$$

For large  $t$  the stretching of the hyper-ellipsoid will be driven by the most positive  $\lambda_i$ . Since the Lyapunov exponents depend weakly on the trajectory, we must actually average on many different points to get an estimate of  $\lambda_i$ :

$$\lambda_i = \lim_{t \rightarrow +\infty} \frac{1}{t} \left[ \lim_{\delta_i\mathbf{x}(0) \rightarrow 0} \ln \left\| \frac{\delta_i\mathbf{x}(t)}{\delta_i\mathbf{x}(0)} \right\| \right] \quad (1.40)$$

The Lyapunov exponents defined so far, are *global* property of the flow: we are often interested in *local* dynamic properties. So we define the leading *Local Lyapunov Vector* (LLV) at time  $t$ : it is the vector  $\mathbf{e}$  towards which converge all random perturbation  $\delta\mathbf{x}(t - \Delta T)$ , started a long time  $\Delta T$  before  $t$ . It may be defined using the Tangent Linear Model:

$$\mathbf{e}(t) = \lim_{\Delta T \rightarrow +\infty} \mathbf{M}(t - \Delta T, t) \delta\mathbf{x}(t - \Delta T) \quad (1.41)$$

After computing the leading local Lyapunov vector, the corresponding local Lyapunov exponent may be computed from the change of its norm.



Another feature which is related to the Lyapunov exponents is the dimensionality of the attractor. The Kaplan-Yorke dimension of the system is defined by:

$$D \equiv k + \frac{\lambda_1 + \lambda_2 + \dots + \lambda_k}{|\lambda_{k+1}|} \quad (1.42)$$

where  $\lambda_1 > \lambda_2 > \dots > \lambda_N$  are the Lyapunov characteristic exponents in decreasing order and  $k$  is the integer for which  $\lambda_1 + \lambda_2 + \dots + \lambda_k > 0$  and  $\lambda_1 + \lambda_2 + \dots + \lambda_k + \lambda_{k+1} < 0$ .

An intuitive justification of eq. 1.42 is the following. The sum of all the exponents is the rate at which the volume of the hyper-ellipsoid will increase or decrease: it will be zero for conservative systems and negative for dissipative ones. If we consider an  $N$ -dimensional box containing the attractor, the sum  $\lambda_1 + \lambda_2 + \dots + \lambda_k$  of the first  $k$  Lyapunov exponents accounts for the rate at which will increase or decrease the  $k$ -dimensional hyper-volume of the projection of an infinitesimal hyper-ellipsoid on the  $k$ -dimensional face of the box. If  $\lambda_1 > 0$  but  $\lambda_1 + \lambda_2 < 0$ , then the projection of the hyper-ellipsoid on one edge of the box will grow, while the projection on a 2-dimensional face will shrink: we may expect the attractor to consist of complex curves, without surfaces.

On the other hand, if  $\lambda_1 + \lambda_2 > 0$  but  $\lambda_1 + \lambda_2 + \lambda_3 < 0$ , then the projection of the hyper-ellipsoid on a 2-dimensional face of the box will grow, while the projection on a 3-dimensional face will shrink: we may expect the attractor to consist of complex surfaces, but no 3-dimensional manifolds. In general, if  $\lambda_1 + \lambda_2 + \dots + \lambda_k > 0$  and  $\lambda_1 + \lambda_2 + \dots + \lambda_k + \lambda_{k+1} < 0$  the attractor may be thought of as consisting of complex  $k$ -dimensional manifolds [19].

A *stable* system will have all Lyapunov exponents less or equal to zero, while a *chaotic* one, whether dissipative or not, will have at least one positive Lyapunov exponent. Furthermore, a *chaotic bounded flow* must have a zero Lyapunov exponent, with the corresponding local Lyapunov vector aligned to the trajectory.

### 1.3.9 Bred vectors

Bred vectors (BVs) represent *finite amplitude* perturbations. Lyapunov vectors (LVs), instead, represent *infinitesimal* perturbations by definition.

The BVs are computed in a similar manner as the LVs, but using the nonlinear model and a finite renormalization amplitude [16]:

- we perturb the vector trajectory  $\mathbf{x}(t)$  with a given finite amplitude random vector  $\delta\mathbf{x}(t)$ ; this random perturbation is introduced only once, at the beginning of the breeding cycle. The size of the initial perturbation is the only tunable parameter of the breeding

procedure, and in operational NWP models it can be used to filter out unwanted fast instabilities, such as convection or even Brownian motion;

- we evolve the resulting perturbed trajectory by using the nonlinear model  $M$ ; the same will be done for the unperturbed trajectory. At fixed time interval  $\Delta t$  we subtract the unperturbed trajectory from the unperturbed one:

$$\delta \mathbf{x}(t + \Delta t) = M(\mathbf{x}(t) + \delta \mathbf{x}(t)) - M(\mathbf{x}(t)) \quad (1.43)$$

where we dropped again the obvious dependence of the nonlinear model  $M$  on  $t$  and  $\Delta t$ .

- we scale down the resulting difference by dividing it by its amplification factor, in order to keep its size the same as the initial one.

As we can see from their definition, BVs are closely related to leading *Local Lyapunov Vectors* (LLVs), since after an infinite breeding time, infinitesimal amplitude bred vectors are identical to LLVs. They share some features, such as the independence on the norm and on the rescaling time, but not others: for example, even without orthogonalization, BVs don't converge to a single leading BV, because of nonlinearity.

### 1.3.10 Dynamical systems described by maps

Also very important are those dynamical systems where the time is a discrete variable. In such a case the dynamical system is described by an  $N$ -dimensional *map*

$$\mathbf{x}_{n+1} = \mathbf{G}(\mathbf{x}_n) \quad (1.44)$$

where  $n$  is the discrete time variable,  $\mathbf{x}_n$  is the  $N$ -dimensional state vector of the system and  $\mathbf{G}$  is the  $N$ -dimensional vector function of the state vector  $\mathbf{x}_n$ ; i.e.  $\mathbf{G}$  evolves the state vector  $\mathbf{x}_n$  at time  $n$ , into the new state vector  $\mathbf{x}_{n+1}$  at time  $n + 1$ . It should be noted that a map can be created by *any* flow, described by eq. 1.2, simply by observing the flow only at regular time intervals.

As an example of a one-dimensional nonlinear map, consider the *logistic map*:

$$x_{n+1} = r x_n (1 - x_n) \quad (1.45)$$

which is a simple population dynamics model akin to the classic logistic equation:

$$\frac{dx}{dt} = r x (1 - x) \quad (1.46)$$

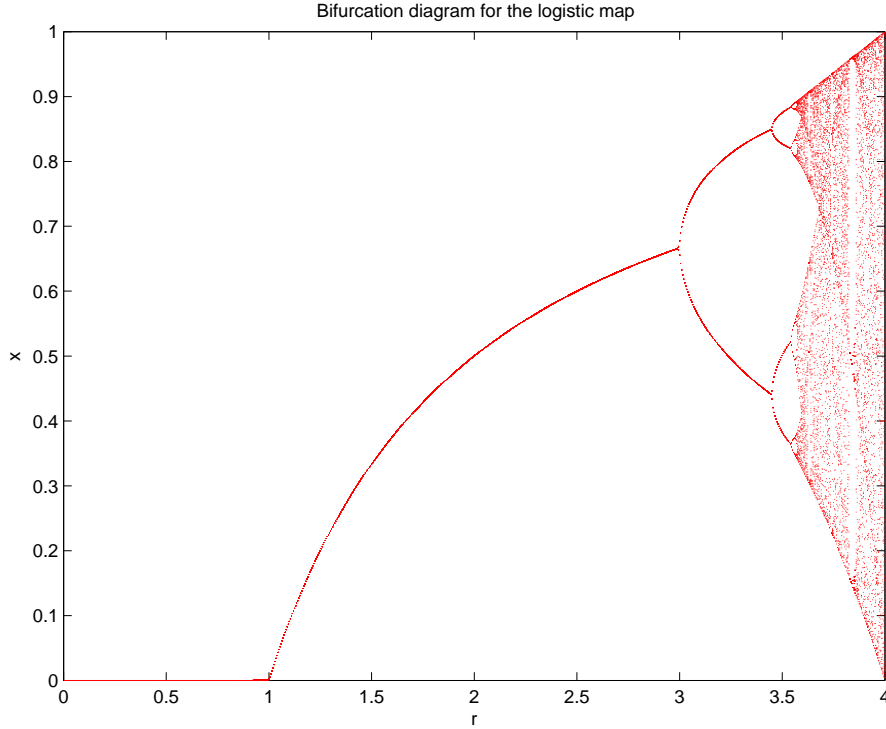


Figure 1.3: Bifurcation diagram for the Logistic map,  $0 \leq r \leq 4$ .

that describes the logistic population growth [22]. Here  $r$  is the intrinsic per capita growth rate, and  $x$  is the population density with respect to the total carrying capacity, i.e.  $0 \leq x \leq 1$ . The logistic map is chaotic with the parameter  $r = 4$ , but it actually displays many different behaviors depending on the value of parameter  $r$ : we can have stable, periodic or chaotic solutions. For example, as we can see in the bifurcation diagram shown in Figures 1.3, 1.4 and 1.5, we have a period 1 solution for  $0 < r < 3$ , while for  $r = 3$  we note the first bifurcation from period 1 to period 2. We observe another bifurcation from period 2 to period 4 when  $r = 1 + \sqrt{6} = 3.45$ . The onset of chaos occurs at  $r \simeq 3.57$ , but there are also other periodic windows, as for example the period 3 window at  $r = 1 + 2\sqrt{2} = 3.83$ . The values of the parameter  $r$  for which we have a periodic behavior is an infinite number of finite intervals, while the values for which it is chaotic, with  $3.57 < r \leq 4$ , form a Cantor set [19].

The bifurcation diagram for the logistic map shown in Figures 1.3, 1.4 and 1.5 — with different zoom windows — have been obtained by plotting as a function of the parameter  $r$  a set of values for  $x_n$  resulting as the evolution of a random value  $x_0$ : we iterated many times, and discarded the first points corresponding to the transient time, before the convergence to the attractor.

The logistic map is *not* invertible, because each points — except the maximum — has not a unique past. Its Lyapunov exponent is  $\lambda = 0.693147 = \ln 2$ .

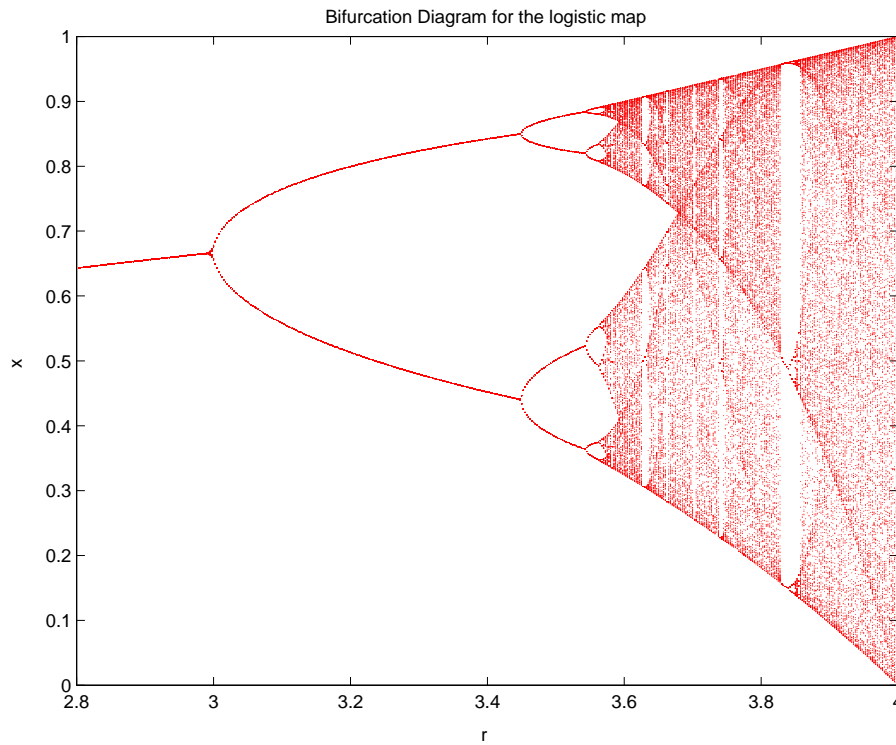


Figure 1.4: Bifurcation diagram for the Logistic map,  $2.8 \leq r \leq 4$ . Notice the period-3 window around  $r = 3.83$ .

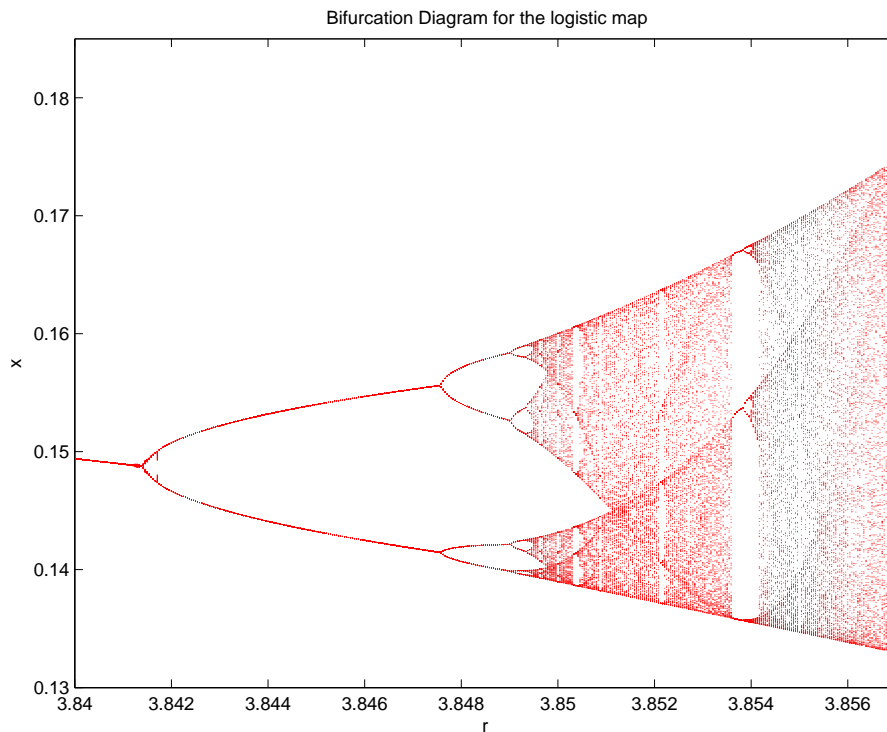
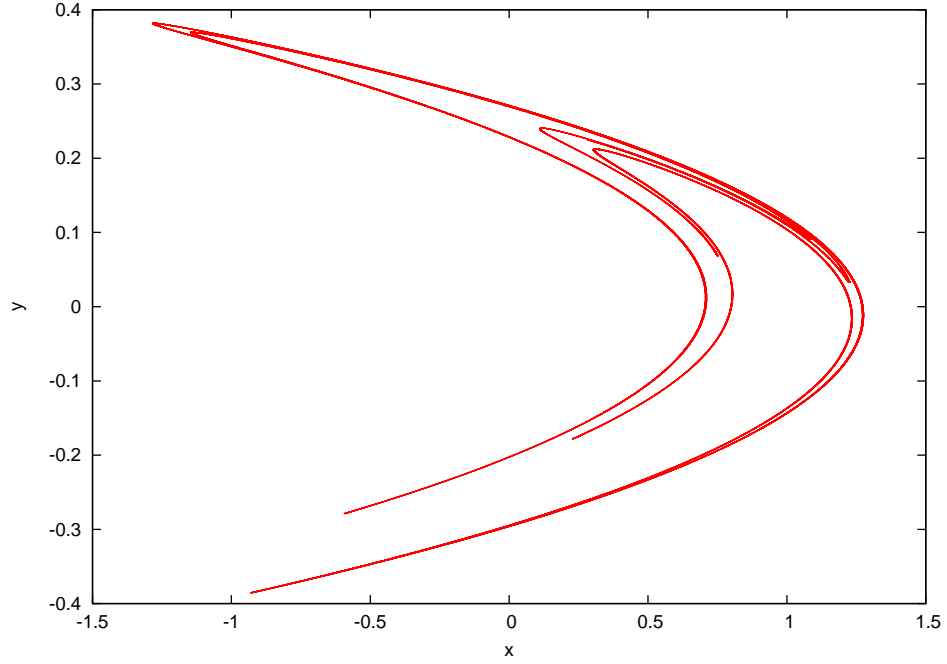


Figure 1.5: Bifurcation diagram for the Logistic map, a zoom. We can observe the self-similarity properties of this diagram.

Figure 1.6: Hénon map: strange attractor for classical parameters  $a$  and  $b$  (see text).



An example of a two-dimensional, dissipative map is the following Hénon map [12][31]:

$$\begin{cases} x_{n+1} &= y_n + 1 - a x_n^2 \\ y_{n+1} &= b x_n \end{cases} \quad (1.47)$$

This map, shown in Fig. 1.6, is also chaotic with the two canonical parameter  $a = 1.4$ ,  $b = 0.3$ . Its attractor is shown below: they have been plotted 100,000 dots after a transient of 10,000 time step, with a starting point  $(x_0, y_0) = (0, 0)$ . The Jacobian  $\mathbf{J}$  of a generic map  $(x_{n+1}, y_{n+1}) = (f(x_n, y_n), g(x_n, y_n))$  is:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial f}{\partial x_n} & \frac{\partial f}{\partial y_n} \\ \frac{\partial g}{\partial x_n} & \frac{\partial g}{\partial y_n} \end{bmatrix} \quad (1.48)$$

For the Hénon map we have  $|\det(\mathbf{J})| < 1$  for all  $x_n$ :

$$|\det(\mathbf{J})| = \left| \det \begin{bmatrix} -2ax_n & 1 \\ b & 0 \end{bmatrix} \right| = |-b| < 1$$

If  $-1 < b < 1$  — as in the case of classic parameter  $b = 0.3$  — this means that the Hénon map is area contracting by a constant factor  $|b|$  for each iteration.

## 1.4 Lorenz's three dimensional chaotic system (1963)

The Lorenz's three dimensional model (1963), hereafter referred to L63, was found by the MIT meteorologist Edward N. Lorenz in early 60s by studying a simplified version of a set of equations modelling a convective fluid motion driven by heating from below.

### 1.4.1 The equations

The flow occurs in a uniform depth layer of fluid, and the temperature difference between upper and lower surfaces has a constant value  $\Delta T$ . The system has a steady-state solution, where no motion occurs and the temperature varies linearly with depth. If, depending on physical conditions, the solution is unstable, a convective motion will arise [18]. In case the motion is completely vertical, with no deviations, if the upper and lower boundaries are taken to be free and an abrupt truncation is performed, the system turns out to simplify in a set of 3 equations only.

Thus, today L63 is known as a dynamical system described by the following dimensionless equations:

$$\left\{ \begin{array}{lcl} \frac{dx}{dt} & = & \sigma(y - x) \\ \frac{dy}{dt} & = & rx - y - xz \\ \frac{dz}{dt} & = & xy - bz \end{array} \right. \quad (1.49)$$

where  $\sigma = 10$ ,  $r = 28$ ,  $b = \frac{8}{3}$  are positive parameters and  $t$  represents the dimensionless time [18]. In these equations the variables  $x$ ,  $y$ ,  $z$  depend on time alone. In our experiments, we will set the integration step  $\Delta t = 0.01$ , and the numerical integration method will be a second order Runge-Kutta scheme (see Appendix A).

A qualitative comparison among three popular numerical integration schemes, (1-st order) Euler, 2-nd order Runge-Kutta and 4-th order Runge-Kutta, is shown in Figures 1.7 and 1.8 for a particular case. The initial condition is on the attractor set and is the same for all schemes:  $(x_0, y_0, z_0) = (14.2041, 15.0165, 34.7172)$ . In both plots the numerical integration is performed 1000 integration steps ahead, with a time step  $\Delta t = 0.01$ . Since they are different order integration schemes, of course they lead to slightly different evolutions of the system. Due to the chaotic dynamics of the system, these differences tend to amplify. A generic initial condition won't be in the attractor set of the system: it will need a transient time to converge to it: so, in practice, in our plots (see Figures 1.9, 1.10, 1.11, 1.12) we set a transient time of

Figure 1.7: Lorenz's 1963 model: a comparison among 1-st order Euler, 2-nd and 4-th order Runge-Kutta schemes for 1000 time steps  $\Delta t = 0.01$ . The initial point belongs to the attractor and is the same for all schemes:  $(x_0, y_0, z_0) = (14.2041, 15.0165, 34.7172)$ .

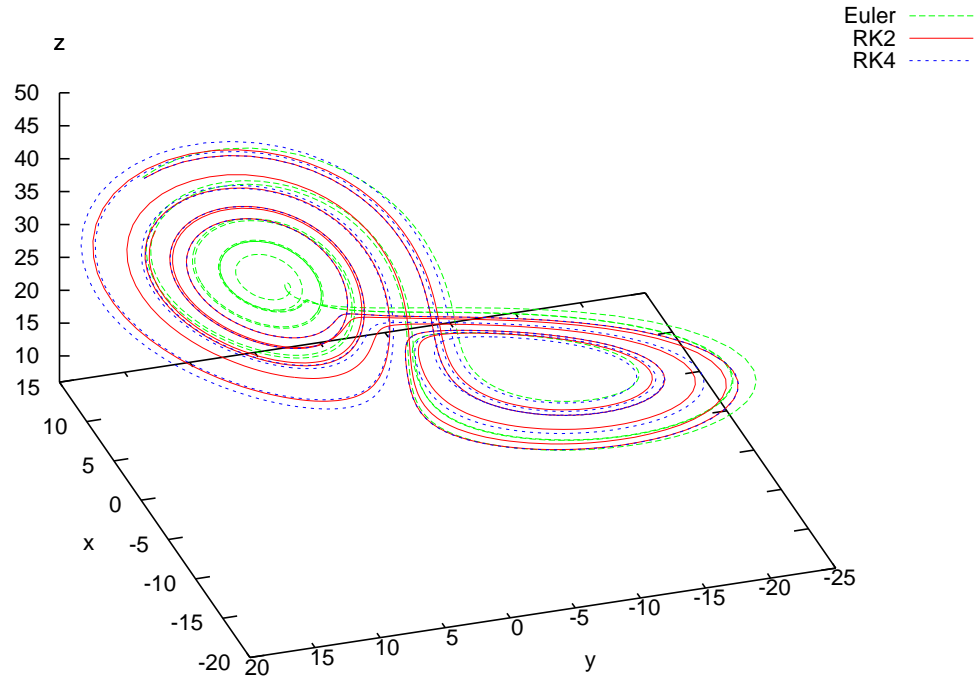


Figure 1.8: Lorenz's 1963 model: a comparison among 1-st order Euler, 2-nd and 4-th order Runge-Kutta schemes for the variable  $x$ . The initial point belongs to the attractor and is the same for all schemes:  $(x_0, y_0, z_0) = (14.2041, 15.0165, 34.7172)$ .

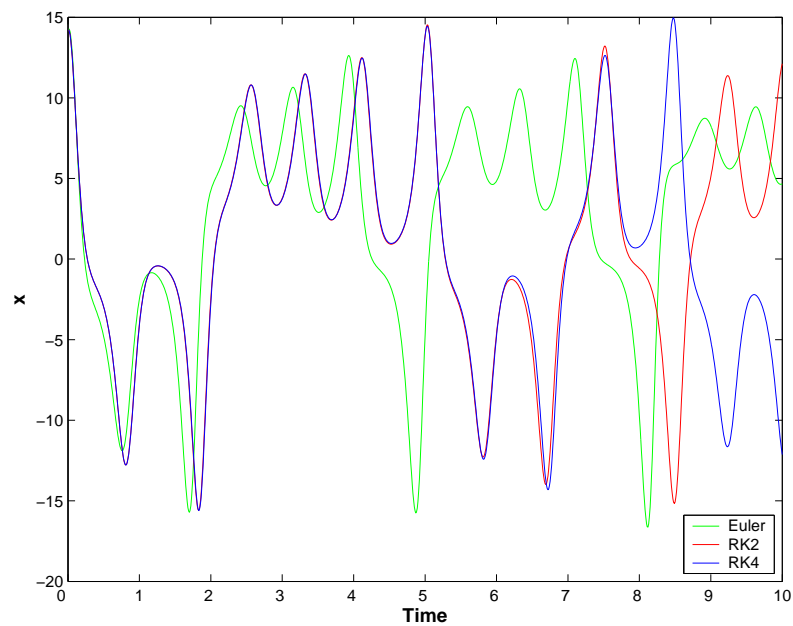
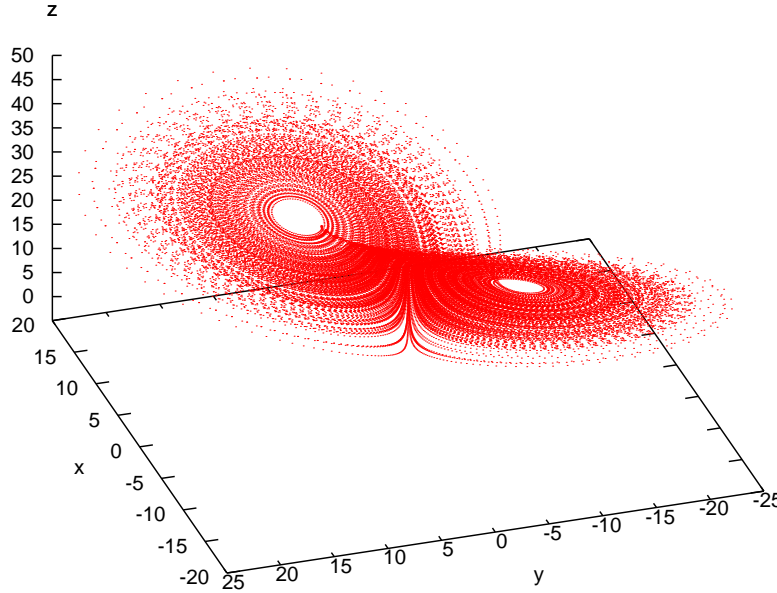


Figure 1.9: Lorenz's 1963 model attractor.



10,000 integration steps.

In Fig. 1.16 we show the so called *Lorenz map*, where the  $(n + 1)$ -th local maximum of  $z_{max}(n + 1)$  is plotted versus  $z_{max}(n)$ , the previous one. See also Fig. 1.15. Note that actually the Lorenz map is not actually a well defined function, since there may be more than one output  $z_{max}(n + 1)$  for an input  $z_{max}(n)$  [31], depending on the other variables  $x$  and  $y$ . So it has a thickness, that prevents us to use this map to deterministically forecast the state of the system. On the other hand it should also be noticed that sometimes there are dots in the map well apart from the great bulk of the other dots. Figure 1.16 has been obtained with a  $50 \times 10^6$  time steps integration after a 10,000 time steps transient, and it contains more than 600,000 dots.

### 1.4.2 The meaning of variables and parameters

In particular, the parameter  $\sigma$  is the *Prandtl number*<sup>4</sup>, and its value is  $\sigma = 10$ , which is typical for cold water or approximately twice that of warm water[30][18];  $r$  is the ratio  $R_a/R_c$  between the *Rayleigh number*<sup>5</sup>  $R_a$  and its critical value  $R_c$ . The critical value  $r$  for instability of steady

<sup>4</sup>The *Prandtl number* is the ratio of viscosity and thermal conductivity of the fluid.

<sup>5</sup>The *Rayleigh number* describes the kind of heat transfer within a fluid: below a critical value for that fluid we have basically conduction; over the critical value convection will set up, and the bulk of heat transfer will be due to it.



Figure 1.10: Lorenz's 1963 model attractor projected on  $xz$  plane.

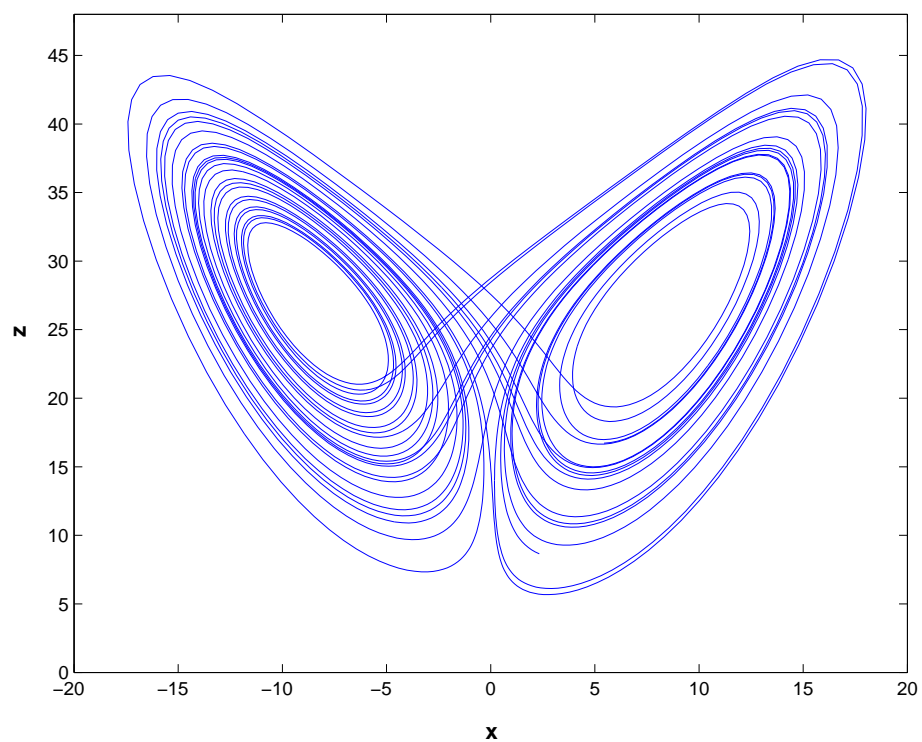


Figure 1.11: Lorenz's 1963 model attractor projected on  $xy$  plane.

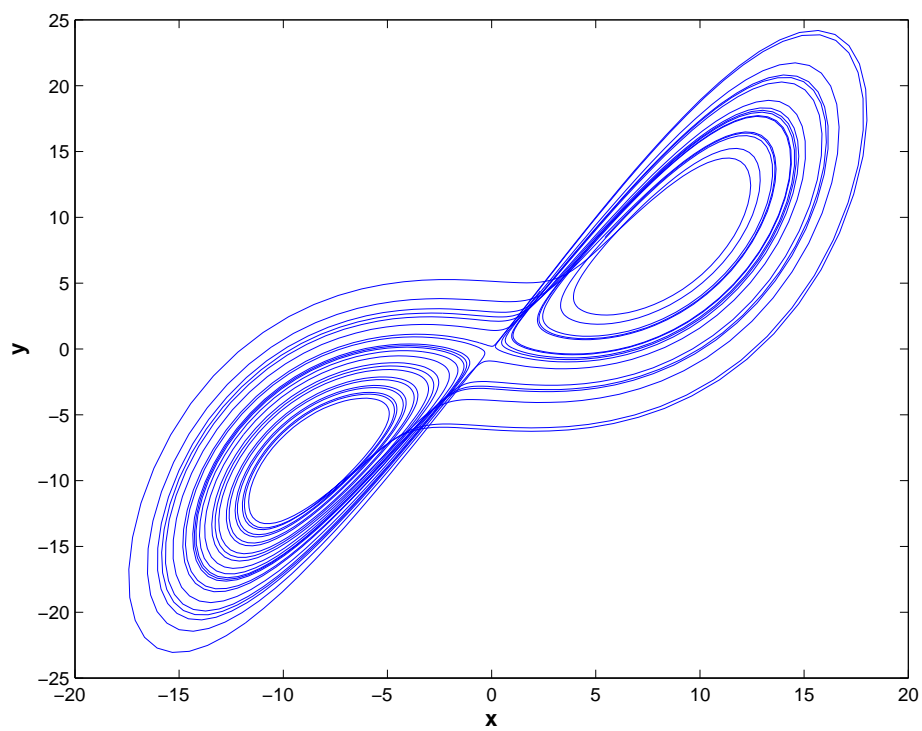


Figure 1.12: Lorenz's 1963 model attractor projected on  $yz$  plane.

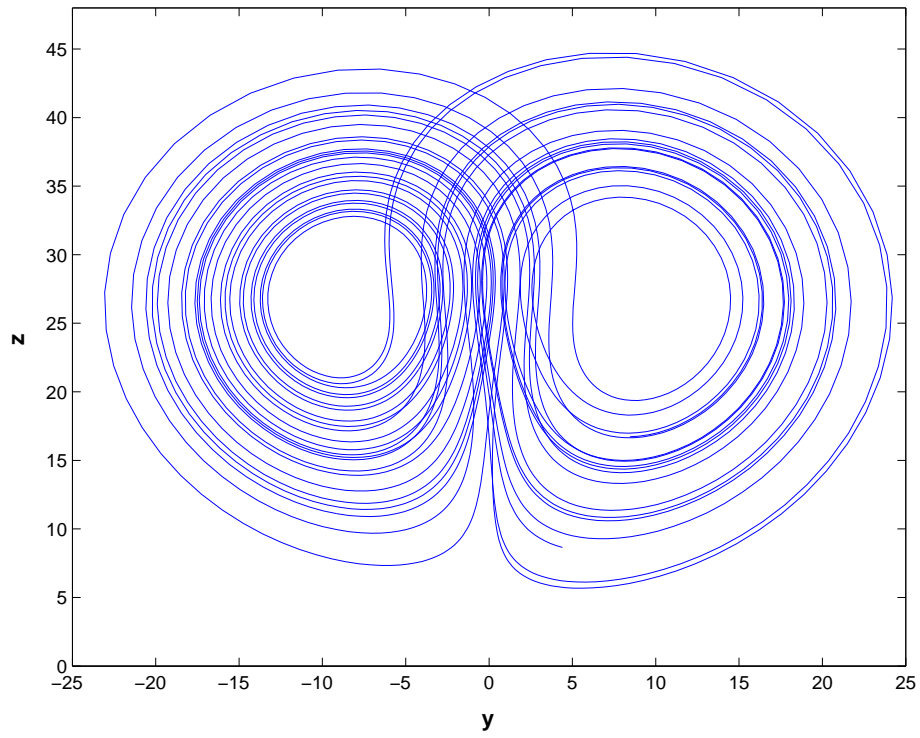


Figure 1.13: Lorenz's 1963 model: time dependence of  $x$ .

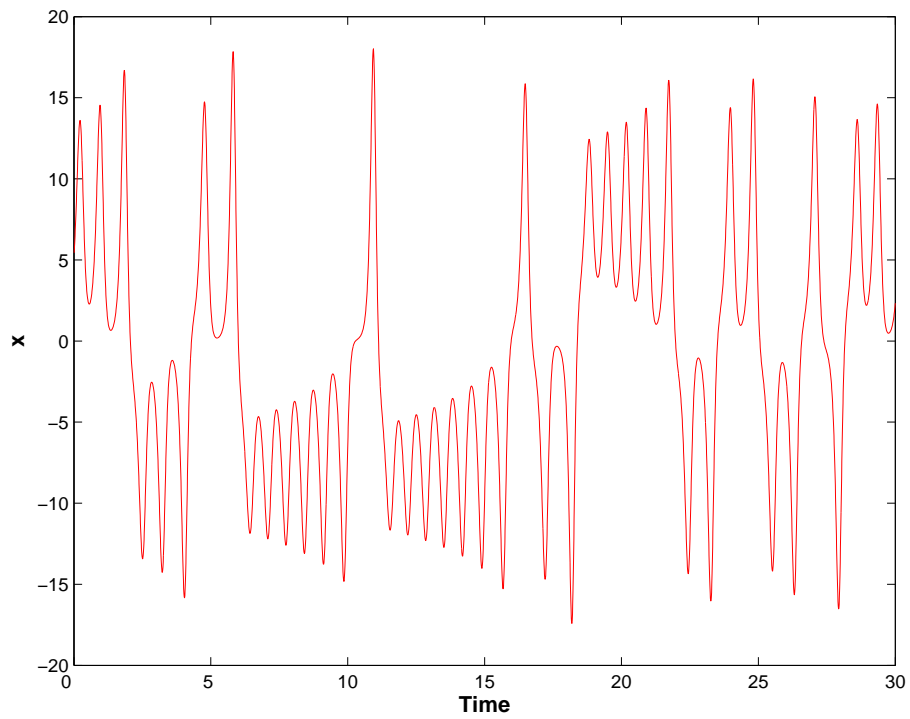


Figure 1.14: Lorenz's 1963 model: time dependence of  $y$ .

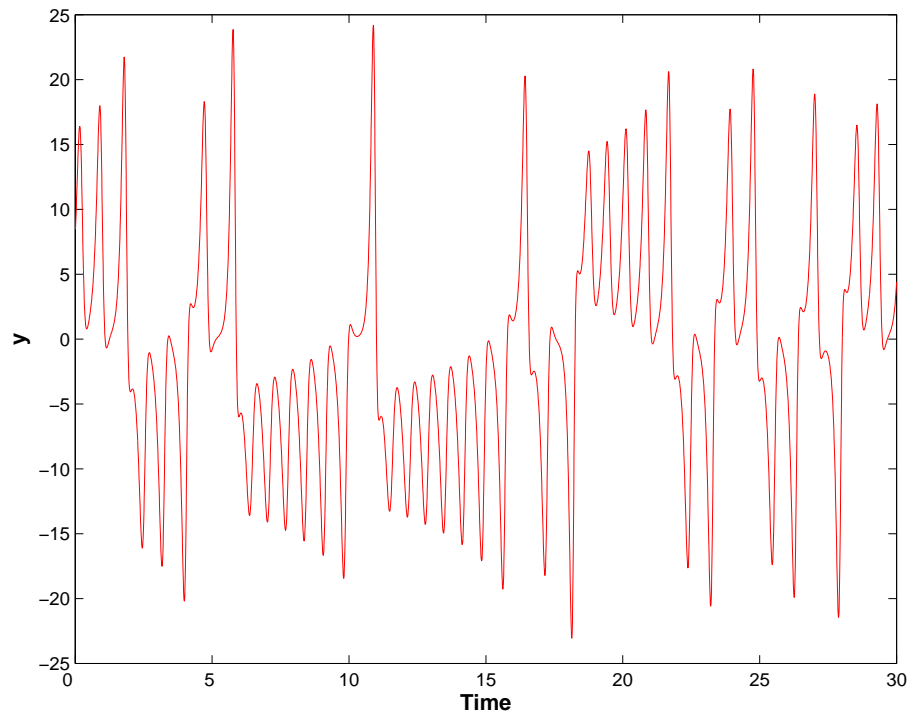


Figure 1.15: Lorenz's 1963 model: time dependence of  $z$ .

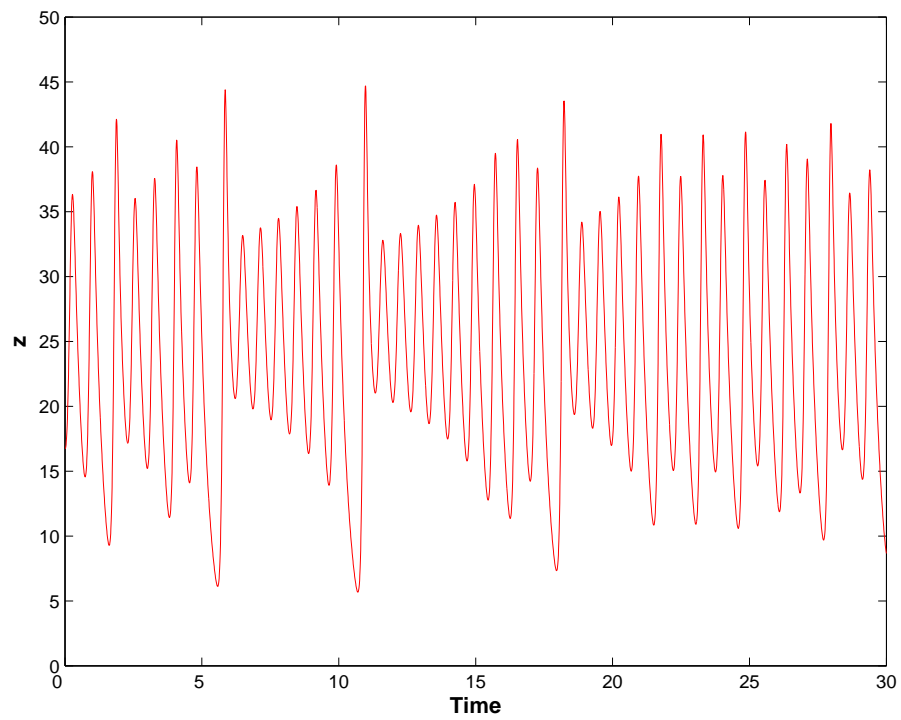
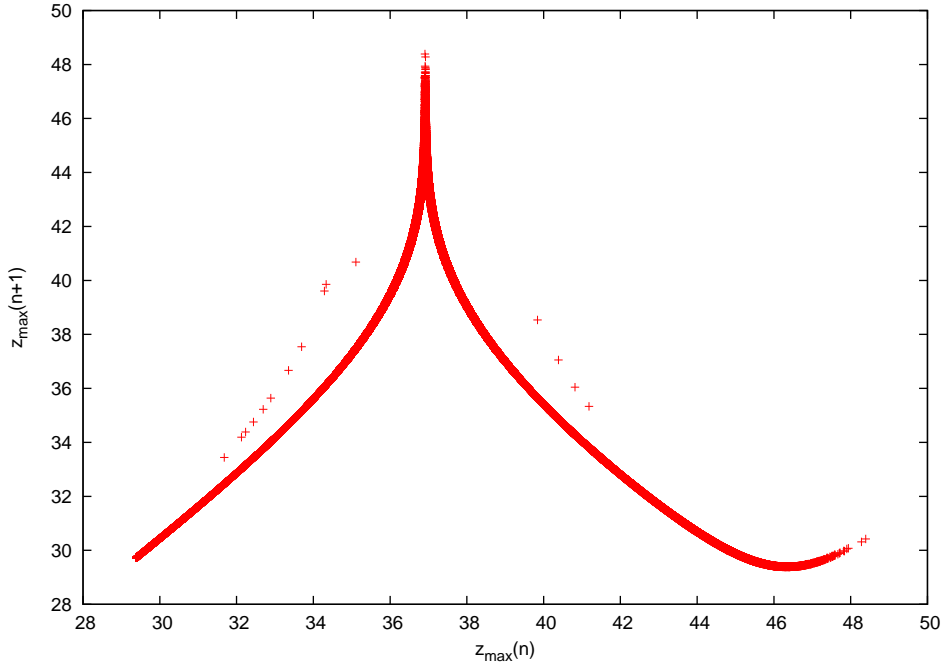


Figure 1.16: Lorenz’s 1963 map: values of the relative maximum  $z_{max}(n)$  and successive relative maximum  $z_{max}(n + 1)$ . See Fig. 1.15.



convection to occur is  $r = 470/19 = 24.74$  [18]. So, a value of  $r = 28$  is slightly supercritical and the flow exhibits unstable convection. The parameter  $b = 8/3$  is proportional to the geometry of the convective cell. For the meaning of the 3 variables  $x$ ,  $y$  and  $z$ , let’s quote Lorenz himself [18]:

“In these equations  $x$  is proportional to the intensity of the convective motion, while  $y$  is proportional to the temperature difference between the ascending and descending currents, similar signs of  $x$  and  $y$  denoting that warm fluid is rising and cold fluid is descending. The variable  $z$  is proportional to the distortion of the vertical temperature profile from linearity, a positive value indicating that the strongest gradients occur near the boundaries.”

Incidentally, note that an initial condition  $x = y = z = 0$  means that the convective system exhibits no convection, and we have a stable “periodic” trajectory.

This system was derived by Lorenz as a drastically simplified model of convection rolls in a fluid heated from below in a gravitational field, but the same equations can also be derived for other physical systems: the equations also exactly describe, for example, the motion of a specific water-wheel [31].

### 1.4.3 Why it is so important: chaos implies limited predictability

Despite its apparent simplicity, the Lorenz's deterministic, nonlinear system exhibits a chaotic dynamics: the solutions oscillate in an irregular way, never exactly repeating themselves and bounded in a particular region of phase space [31, 26]. There are no analytic solutions of the system, so the solutions have to be found by numerical integration only. Lorenz felt that *the really important finding was the fact that under fairly general conditions a lack of periodicity implied limited predictability* [19]. We deal with this problem even for deterministic flows such as, for example, the NWP models. *A fortiori* the atmosphere itself has a finite horizon of predictability, which Lorenz estimated to be approximately two weeks. That is due to deep, dynamical reasons — the chaotic behavior of the system — and cannot be overcome by increased power of computational devices.

### 1.4.4 Lyapunov exponents, dimensionality and doubling time

Lorenz's system L63 is a nonlinear but autonomous system: in eq. 1.2 the function  $\mathbf{F}$  does not depend explicitly on time. L63 is also a *dissipative, nonconservative* system, i.e. the time evolution does not preserve volumes in phase space. In general, a volume  $V(t)$  of a phase space region  $D$  shrinks according to 1.13:

$$\left. \frac{dV}{dt} \right|_{t=0} = \int_D \nabla \cdot \mathbf{F}(\mathbf{x}) d\mathbf{x}$$

For L63 the divergence of the flow is:

$$\nabla \cdot \mathbf{F} = -(\sigma + 1 + b) \quad (1.50)$$

and an initial volume  $V(0)$  in phase space will shrink according to eq. 1.18:

$$V(t) = V(0) \exp[-t(\sigma + 1 + b)] \quad (1.51)$$

$$= V(0) \exp(-13.67 t) \quad (1.52)$$

The Lorenz's system has three Lyapunov exponents, which drive its dynamical features:

$$\lambda_1 = 0.9056 \quad (1.53)$$

$$\lambda_2 = 0 \quad (1.54)$$

$$\lambda_3 = -14.5723 \quad (1.55)$$

Since  $\lambda_1 + \lambda_2 > 0$  and  $\lambda_1 + \lambda_2 + \lambda_3 < 0$ , the Kaplan-Yorke dimension of the attractor of the system (eq. 1.42) is:

$$D = 2 + \frac{\lambda_1 + \lambda_2}{|\lambda_3|} \quad (1.56)$$

$$= 2.06215 \quad (1.57)$$

The *doubling time* is the average time after which a small perturbation will double. It may be calculated from eq. 1.36 coupled with the largest Lyapunov exponent  $\lambda_1$  in the following way:

$$2 \|\delta_0\| = \|\delta_0\| e^{\lambda_1 \tau_{double}} \quad (1.58)$$

$$\implies \lambda_1 \tau_{double} = \ln 2 \quad (1.59)$$

$$\implies \tau_{double} = \frac{\ln 2}{0.9065} = 0.7646 \quad (1.60)$$

## 1.5 The aim of this study

As we stated in section 1.2, the goal of Data Assimilation is to produce an optimal estimate of the state of a dynamical system from incomplete, noisy observations and (approximate) knowledge of the laws governing the evolution of the system. An important application is the initialization of forecast models of the Atmosphere and Ocean.

In this work we address the problem of data assimilation in chaotic systems in the framework of Lorenz's three variable convective model. The aim is to investigate the theory for an advanced formulation of the Assimilation in the Unstable Subspace (AUS, see section 2.3), a data assimilation scheme relying upon the system's dynamics which is applicable to different contexts and computationally affordable in operational environments. AUS has already provided encouraging results with realistic models and observational configurations. The low dimensionality of the model used here allows for the direct comparison with a "golden standard" data assimilation scheme, the Extended Kalman Filter, and for development of the theory.

## 1.6 Notation and conventions

Throughout this study and regardless to the scheme under consideration, we will conform to the notation proposed by Ide *et al.* [14], widely used by the scientific community for data assimilation.

### 1.6.1 Model space and observational space

The evolution of a dynamical discrete system from time  $t_k$  to time  $t_{k+1}$  is described by the equation

$$\mathbf{x}^f(t_{k+1}) = M_k [\mathbf{x}^a(t_k)] \quad (1.61)$$

where the column vector  $\mathbf{x}^f(t_{k+1})$  is the *forecast state* at time  $t_{k+1}$  and the column vector  $\mathbf{x}^a(t_k)$  is the *analysis state* at time  $t_k$ . Both have dimension  $N$  and are defined in the *model space*  $S^{Mod}$ : if the system is described by  $K$  variables at  $L$  locations, then  $N = K \times L$ . If we have  $M$  observations of the state of the system, we can define the *observations column vector*  $\mathbf{y}^o$ , and its *observational space*  $S^{ob}$ , i.e.:  $\mathbf{y}^o \in S^{ob}$ . In an operational context, in general the available observations are much less than the dimensionality of the NWP model, so  $M \ll N$ . The assimilation algorithm, aimed at initializing our model with the analysis state  $\mathbf{x}^a$ , can be thought of as an application  $F$ , that combines both the forecast state  $\mathbf{x}^f$  and the observations  $\mathbf{y}^o$  to provide the analysis state  $\mathbf{x}^a$ :

$$F : S^{Mod} \times S^{ob} \longrightarrow S^{Mod} \quad (1.62)$$

$$(\mathbf{x}^f, \mathbf{y}^o) \longmapsto \mathbf{x}^a \quad (1.63)$$

In this work we will follow the common practice introduced by Rutherford in 1972 [14], who proposed  $\mathbf{x}^b = \mathbf{x}^f$ , since forecast states had become better background fields than climatology. Other choices may be taken for the vector field  $\mathbf{x}^b$ , for example by averaging over an ensemble of different forecasts.

### 1.6.2 The observation operator $H$ and its linear approximation $\mathbf{H}$

The *observation operator*  $H$  is defined in the model space  $S^{Mod}$ :

$$H : S^{Mod} \longrightarrow S^{ob} \quad (1.64)$$

It transforms an  $N$ -dimensional state vector  $\mathbf{x} \in S^{Mod}$  into an “observational”  $M$ -dimensional vector. Its structure contains all mathematical and physical relations allowing for an *a priori* estimate of the observations: of course, in general  $H$  is *not* linear. But the nonlinear operator  $H$  can be linearized around the trajectory  $\mathbf{x}$ , which we assume it’s well approximated by our forecast trajectory  $\mathbf{x}^f$ , with

$$H(\mathbf{x} + \delta\mathbf{x}) = H(\mathbf{x}) + \mathbf{H} \delta\mathbf{x} \quad (1.65)$$

where  $\delta \mathbf{x}$  is a small perturbation of the state and  $\mathbf{H}$  may be represented by an  $M \times N$  matrix, whose elements are [16]:

$$(\mathbf{H})_{ij} = \frac{\partial H_i}{\partial x_j} \quad \text{with } i \in \{1, \dots, M\}, j \in \{1, \dots, N\} \quad (1.66)$$

So the operator  $\mathbf{H}$  is the Jacobian of the operator  $H$ , and transforms vectors in model space into their corresponding vectors in observation space. Its transpose  $\mathbf{H}^T$  transforms vectors in observation space into vectors in model space.

Note that, if the operator  $H$  is linear, by definition of linearity it holds:

$$H(\mathbf{x} + \delta \mathbf{x}) = H(\mathbf{x}) + H \delta \mathbf{x} \quad (1.67)$$

and, by using eq. 1.65, we have

$$H = \mathbf{H} \quad (1.68)$$

### 1.6.3 Error vectors

The model operator  $M_k$  in eq. 1.61 evolves the state from time  $t_k$  to time  $t_{k+1}$ . If  $\mathbf{x}^t(t_k)$  is the *true state* of the system at time  $t_k$ , the corresponding *analysis error* and the *forecast error* will be the column vectors

$$\boldsymbol{\eta}^a(t_k) = \mathbf{x}^a(t_k) - \mathbf{x}^t(t_k) \quad (1.69)$$

$$\boldsymbol{\eta}^f(t_k) = \mathbf{x}^f(t_k) - \mathbf{x}^t(t_k) \quad (1.70)$$

These errors are defined in the model space  $S^{Mod}$ .

Generally speaking, even the observation operator  $H$  may be misrepresented:

$$\boldsymbol{\eta}^H(t_k) = \mathbf{y}^t(t_k) - H\mathbf{x}^t(t_k) \quad (1.71)$$

where  $\mathbf{y}^t(t_k)$  is the vector of true values of the observed variables at time  $t_k$ . The vector  $\boldsymbol{\eta}^H$  is defined in the observational space  $S^{ob}$ , as well the *observational error*:

$$\boldsymbol{\eta}^o(t_k) = \mathbf{y}^o(t_k) - H\mathbf{x}^t(t_k) \quad (1.72)$$

The observational error  $\boldsymbol{\eta}^o$  implicitly includes the measurements error, the observation operator misrepresentation and the model's *error of representativeness* due to subgrid-scale processes not represented in our grid-averaged values of the model and analysis [16].



The model error between the times  $t_k$  and  $t_{k+1}$  is defined by

$$\boldsymbol{\eta}^M(t_k) = M_k [\mathbf{x}^t(t_k)] - \mathbf{x}^t(t_{k+1}) \quad (1.73)$$

Rearranging it with eq. 1.61,  $\mathbf{x}^f(t_{k+1}) = M_k [\mathbf{x}^a(t_k)]$ , we have:

$$\mathbf{x}^f(t_{k+1}) - \mathbf{x}^t(t_{k+1}) = M_k [\mathbf{x}^a(t_k)] - M_k [\mathbf{x}^t(t_k)] + \boldsymbol{\eta}^M(t_k) \quad (1.74)$$

The lhs is the forecast error at time  $t_{k+1}$ . So, assuming  $M_k [\mathbf{x}^a(t_k)] - M_k [\mathbf{x}^t(t_k)]$  to be a small term, we can calculate the new forecast error with the tangent linear model  $\mathbf{M}_k$ :

$$\boldsymbol{\eta}^f(t_{k+1}) = \mathbf{M}_k \boldsymbol{\eta}^a(t_k) + \boldsymbol{\eta}^M(t_k) \quad (1.75)$$

#### 1.6.4 Error covariance matrices

Now we can define from the above column vectors their covariance matrices, built by right-multiplying each of them by its transpose and taking the expectation values. In practice, the expectation values are estimated by averaging on many cases. The analysis error covariance matrix  $\mathbf{P}^a$ , the forecast error covariance matrix  $\mathbf{P}^f$  and the observation error covariance matrix  $\mathbf{R}$  then read:

$$\mathbf{P}^a = \langle \boldsymbol{\eta}^a (\boldsymbol{\eta}^a)^T \rangle \quad (1.76)$$

$$\mathbf{P}^f = \langle \boldsymbol{\eta}^f (\boldsymbol{\eta}^f)^T \rangle \quad (1.77)$$

$$\mathbf{R} = \langle \boldsymbol{\eta}^o (\boldsymbol{\eta}^o)^T \rangle \quad (1.78)$$

More explicitly:

$$\mathbf{P}^a = \begin{bmatrix} \langle \eta_1^a \eta_1^a \rangle & \langle \eta_1^a \eta_2^a \rangle & \cdots & \langle \eta_1^a \eta_N^a \rangle \\ \langle \eta_2^a \eta_1^a \rangle & \langle \eta_2^a \eta_2^a \rangle & \cdots & \langle \eta_2^a \eta_N^a \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \eta_N^a \eta_1^a \rangle & \langle \eta_N^a \eta_2^a \rangle & \cdots & \langle \eta_N^a \eta_N^a \rangle \end{bmatrix} \quad (1.79)$$

$$\mathbf{P}^f = \begin{bmatrix} \langle \eta_1^f \eta_1^f \rangle & \langle \eta_1^f \eta_2^f \rangle & \cdots & \langle \eta_1^f \eta_N^f \rangle \\ \langle \eta_2^f \eta_1^f \rangle & \langle \eta_2^f \eta_2^f \rangle & \cdots & \langle \eta_2^f \eta_N^f \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \eta_N^f \eta_1^f \rangle & \langle \eta_N^f \eta_2^f \rangle & \cdots & \langle \eta_N^f \eta_N^f \rangle \end{bmatrix} \quad (1.80)$$

$$\mathbf{R} = \begin{bmatrix} \langle \eta_1^o \eta_1^o \rangle & \langle \eta_1^o \eta_2^o \rangle & \cdots & \langle \eta_1^o \eta_M^o \rangle \\ \langle \eta_2^o \eta_1^o \rangle & \langle \eta_2^o \eta_2^o \rangle & \cdots & \langle \eta_2^o \eta_M^o \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \eta_M^o \eta_1^o \rangle & \langle \eta_M^o \eta_2^o \rangle & \cdots & \langle \eta_M^o \eta_M^o \rangle \end{bmatrix} \quad (1.81)$$

Since each vector depends on time  $t_k$  all the above error covariance matrices depend on time  $t_k$  as well. Note also that  $\mathbf{P}^a$  and  $\mathbf{P}^f$  are  $N \times N$  matrices, while  $\mathbf{R}$  is an  $M \times M$  one. Note that the observation error covariance matrix  $\mathbf{R}$  can be actually thought of as including three components [16]: instrument error covariance matrix  $\mathbf{R}_{instr}$ , the representativeness error covariance matrix  $\mathbf{R}_{repr}$ , both assumed to be uncorrelated, and the observation operator  $H$  error covariance matrix  $\mathbf{R}_H$ :

$$\mathbf{R} = \mathbf{R}_{instr} + \mathbf{R}_{repr} + \mathbf{R}_H \quad (1.82)$$

From eq. 1.75,  $\boldsymbol{\eta}^f(t_{k+1}) = \mathbf{M}_k \boldsymbol{\eta}^a(t_k) + \boldsymbol{\eta}^M(t_k)$ , we have [14]:

$$\mathbf{P}_{k+1}^f = [\mathbf{M}_k \boldsymbol{\eta}^a(t_k) + \boldsymbol{\eta}^M(t_k)] [\mathbf{M}_k \boldsymbol{\eta}^a(t_k) + \boldsymbol{\eta}^M(t_k)]^T \quad (1.83)$$

$$= \mathbf{M}_k \mathbf{P}_k^a \mathbf{M}_k^T + \mathbf{Q}_k \quad (1.84)$$

where the term  $\mathbf{Q}_k$  refers to the model error covariance matrix between the times  $t_k$  and  $t_{k+1}$ . It is defined, in a similar way as  $\mathbf{P}^a$ ,  $\mathbf{P}^f$  and  $\mathbf{R}$ , by the model error  $\boldsymbol{\eta}^M(t_k)$ :

$$\mathbf{Q} = \langle \boldsymbol{\eta}^M (\boldsymbol{\eta}^M)^T \rangle \quad (1.85)$$

(here we dropped the subscript  $k$ ). That is:

$$\mathbf{Q} = \begin{bmatrix} \langle \eta_1^M \eta_1^M \rangle & \langle \eta_1^M \eta_2^M \rangle & \cdots & \langle \eta_1^M \eta_N^M \rangle \\ \langle \eta_2^M \eta_1^M \rangle & \langle \eta_2^M \eta_2^M \rangle & \cdots & \langle \eta_2^M \eta_N^M \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \eta_N^M \eta_1^M \rangle & \langle \eta_N^M \eta_2^M \rangle & \cdots & \langle \eta_N^M \eta_N^M \rangle \end{bmatrix} \quad (1.86)$$

### 1.6.5 Operators and vectors: a low dimensional example

As an example, consider a 3-dimensional space model  $S^{Mod}$  and a 2-dimensional observation space  $S^{ob}$ : the truth vector is  $3 \times 1$ , with components expressed in term of the grid points  $e, f, g$  (see Fig. 1.17):

$$\mathbf{x}^t = \begin{bmatrix} x_e^t \\ x_f^t \\ x_g^t \end{bmatrix} \quad (1.87)$$

and similarly for analysis, background and forecast column vectors:

$$\mathbf{x}^a = [x_e^a \ x_f^a \ x_g^a]^T \quad (1.88)$$

$$\mathbf{x}^b = [x_e^b \ x_f^b \ x_g^b]^T \quad (1.89)$$

$$\mathbf{x}^f = [x_e^f \ x_f^f \ x_g^f]^T \quad (1.90)$$

The observation vector, instead, is a  $2 \times 1$  one expressed in term of the observation points 1 and 2 (see Fig. 1.17 again):

$$\mathbf{y}^o = \begin{bmatrix} y_1^o \\ y_2^o \end{bmatrix} \quad (1.91)$$

The analysis and forecast covariance matrices are:

$$\mathbf{P}^a = \begin{bmatrix} p_{ee}^a & p_{ef}^a & p_{eg}^a \\ p_{fe}^a & p_{ff}^a & p_{fg}^a \\ p_{ge}^a & p_{gf}^a & p_{gg}^a \end{bmatrix} \quad (1.92)$$

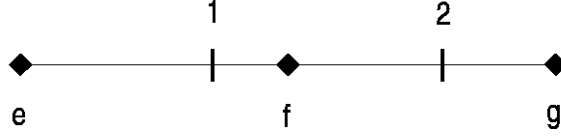
$$\mathbf{P}^f = \begin{bmatrix} p_{ee}^f & p_{ef}^f & p_{eg}^f \\ p_{fe}^f & p_{ff}^f & p_{fg}^f \\ p_{ge}^f & p_{gf}^f & p_{gg}^f \end{bmatrix} \quad (1.93)$$

while the observation error covariance matrix will be:

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} \quad (1.94)$$

If measurements error at different locations are uncorrelated,  $\mathbf{R}$  will be a diagonal matrix. The linearized observation operator  $\mathbf{H}$  projects vectors in model space  $S^{Mod}$  (grid points) into vectors in observation space  $S^{ob}$  (observation points). Its components may be simple

Figure 1.17: A simple case: 3 grid points (e, f, g) and 2 observations (1, 2).



interpolation coefficients:

$$\mathbf{H} = \begin{bmatrix} h_{1e} & h_{1f} & h_{1g} \\ h_{2e} & h_{2f} & h_{2g} \end{bmatrix} \quad (1.95)$$

So for example the background values at observation points are the following vector  $\mathbf{y}^b$ :

$$\mathbf{H}\mathbf{x}^b = \begin{bmatrix} h_{1e} & h_{1f} & h_{1g} \\ h_{2e} & h_{2f} & h_{2g} \end{bmatrix} \begin{bmatrix} x_e^b \\ x_f^b \\ x_g^b \end{bmatrix} \quad (1.96)$$

$$= \begin{bmatrix} y_1^b \\ y_2^b \end{bmatrix} = \mathbf{y}^b \quad (1.97)$$

In Kalman Filter (see subsection 2.2.1) we will also have the term:

$$\mathbf{P}^f \mathbf{H}^T = \begin{bmatrix} p_{ee}^f & p_{ef}^f & p_{eg}^f \\ p_{fe}^f & p_{ff}^f & p_{fg}^f \\ p_{ge}^f & p_{gf}^f & p_{gg}^f \end{bmatrix} \begin{bmatrix} h_{1e} & h_{2e} \\ h_{1f} & h_{2f} \\ h_{1g} & h_{2g} \end{bmatrix} = \begin{bmatrix} p_{e1}^f & p_{e2}^f \\ p_{f1}^f & p_{f2}^f \\ p_{g1}^f & p_{g2}^f \end{bmatrix} \quad (1.98)$$

that is a *grid to observation points* approximation, by interpolation, of the forecast error covariance matrix  $\mathbf{P}^f$ ; as for example in  $p_{e2}^f = p_{ee}^f h_{2e} + p_{ef}^f h_{2f} + p_{eg}^f h_{2g}$ . We will also see the term:

$$\mathbf{H} \mathbf{P}^f \mathbf{H}^T = \begin{bmatrix} h_{1e} & h_{1f} & h_{1g} \\ h_{2e} & h_{2f} & h_{2g} \end{bmatrix} \begin{bmatrix} p_{ee}^f & p_{ef}^f & p_{eg}^f \\ p_{fe}^f & p_{ff}^f & p_{fg}^f \\ p_{ge}^f & p_{gf}^f & p_{gg}^f \end{bmatrix} \begin{bmatrix} h_{1e} & h_{2e} \\ h_{1f} & h_{2f} \\ h_{1g} & h_{2g} \end{bmatrix} = \begin{bmatrix} p_{11}^f & p_{12}^f \\ p_{12}^f & p_{22}^f \end{bmatrix} \quad (1.99)$$

which in turn is an approximation by back interpolation of the forecast error covariance matrix  $\mathbf{P}^f$  between observation points [16].

We will also find the gain matrix

$$\mathbf{K} = \mathbf{P}^f \mathbf{H}^T \left[ \mathbf{H} \mathbf{P}^f \mathbf{H}^T + \mathbf{R} \right]^{-1} \quad (1.100)$$

that is an  $N \times M$  matrix. In this low dimensional example it's  $3 \times 2$ :

$$\mathbf{K} = \begin{bmatrix} p_{e1}^f & p_{e2}^f \\ p_{f1}^f & p_{f2}^f \\ p_{g1}^f & p_{g2}^f \end{bmatrix} \left\{ \begin{bmatrix} p_{11}^f & p_{21}^f \\ p_{12}^f & p_{22}^f \end{bmatrix} + \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} \right\}^{-1} \quad (1.101)$$



## Chapter 2

# Data Assimilation: state of the art

A NWP forecast is an initial value problem coupled with a boundary problem: carefully initialized and with the appropriate boundary conditions, a NWP model outputs the atmospheric evolution (forecast). Of course, to improve the quality of the forecast we need to better estimate the initial conditions, or the present state of the system. As we already stated in paragraph 1.2, the purpose of Data Assimilation in NWP models is “using all the available information, to determine as accurately as possible the state of the atmospheric (or oceanic) flow” (Talagrand, 1997, [16]). The available information is a *statistical combination of (noisy) observations and a short-range forecast*.

In this chapter we will survey the most important techniques devised to combine these informations, without any attempt to be exhaustive. In particular, we will talk about *Variational* and *Sequential* assimilation techniques. The former has been applied since the early 70s, greatly enhanced at the end of the 80s, and still largely used in operational contexts. The latter have an appealing probabilistic approach, but their implementation in realistic geophysical models turns out to be problematic, for reasons to be discussed below.

### 2.1 Variational assimilation

In the variational method basically the problem is to find a model trajectory which best fit the observational data within a given time interval  $\tau$ , often called *assimilation window*, while satisfying the dynamical constraints: in the *strong constraints formulation* the constraints have to be satisfied exactly; in the *weak constraints formulation* they have to be satisfied only ap-

proximately [8]. Since the model equations are deterministic, all we need is the initial state and the boundary conditions, and the variational problem may be restated from a constrained to an unconstrained form. Thus, the idea is to find the initial state  $\mathbf{x}(t_0)$  which evolution under the model equations best fit the observational data within the assimilation window.

In practice, we are looking for the initial state  $\mathbf{x}(t_0)$  that minimizes a so called *cost function*, which is a scalar functional of the trajectory  $\mathbf{x}(t)$ , to be minimized under the model equations constraint 1.2. The cost function  $J(\mathbf{x}(t))$  is defined as the sum of the squares of the distance between  $\mathbf{x}(t)$  and the forecast state  $\mathbf{x}^f(t)$  weighted by the inverse of the forecast error covariance matrix  $\mathbf{P}^f(t)$ , plus the distance between the “first guess” observations  $H\mathbf{x}(t)$  and the true observations  $\mathbf{y}^o(t)$ , weighted by the inverse of the observations error covariance matrix  $\mathbf{R}$  [16].

$$J(\mathbf{x}(t)) = \frac{1}{2} [\mathbf{x}(t) - \mathbf{x}^f(t)]^T (\mathbf{P}^f(t))^{-1} [\mathbf{x}(t) - \mathbf{x}^f(t)] + \frac{1}{2} [H\mathbf{x}(t) - \mathbf{y}^o(t)]^T \mathbf{R}^{-1} [H\mathbf{x}(t) - \mathbf{y}^o(t)] \quad (2.1)$$

### 2.1.1 Maximum likelihood approach

Let's see a *maximum likelihood* motivation for eq. 2.1. Consider first, as an illustrative example, a 1-dimensional problem: we have two independent temperature observations  $T_1$  and  $T_2$ , both assumed to have normally distributed errors with standard deviations  $\sigma_1$  and  $\sigma_2$ . The analysis temperature  $T$  is *the most likely value*, given the two observations  $T_1$  and  $T_2$  and their statistical errors. The cost function, in this particular case, is:

$$J(T) = \frac{1}{2} \left[ \frac{(T - T_1)^2}{\sigma_1^2} + \frac{(T - T_2)^2}{\sigma_2^2} \right] \quad (2.2)$$

The probability distribution of an observation  $T_1$  given a true value  $T$  and a standard deviation  $\sigma_1$  for  $T_1$  is

$$p_{\sigma_1}(T_1|T) = \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(T_1-T)^2}{2\sigma_1^2}} \quad (2.3)$$

A similar relation holds for  $T_2$ :

$$p_{\sigma_2}(T_2|T) = \frac{1}{\sqrt{2\pi}\sigma_2} e^{-\frac{(T_2-T)^2}{2\sigma_2^2}} \quad (2.4)$$

The likelihood of a true value  $T$  given an observation  $T_1$  with a standard deviation  $\sigma_1$  is given by [16]:

$$L_{\sigma_1}(T|T_1) = p_{\sigma_1}(T_1|T) = \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(T_1-T)^2}{2\sigma_1^2}} \quad (2.5)$$

and — in the same way — the likelihood of a true value  $T$  given an observation  $T_2$  with a



standard deviation  $\sigma_2$  is:

$$L_{\sigma_2}(T|T_2) = p_{\sigma_2}(T_2|T) = \frac{1}{\sqrt{2\pi}\sigma_2} e^{-\frac{(T_2-T)^2}{2\sigma_2^2}} \quad (2.6)$$

Since the two observations  $T_1$  and  $T_2$  are independent, the likelihood of a true value  $T$  given *both*  $T_1$  and  $T_2$  is the product:

$$L_{\sigma_1\sigma_2}(T|T_1, T_2) = L_{\sigma_1}(T|T_1) \cdot L_{\sigma_2}(T|T_2) \quad (2.7)$$

$$\begin{aligned} &= \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(T_1-T)^2}{2\sigma_1^2}} \frac{1}{\sqrt{2\pi}\sigma_2} e^{-\frac{(T_2-T)^2}{2\sigma_2^2}} \\ &= \frac{1}{2\pi\sigma_1\sigma_2} e^{-\frac{(T_1-T)^2}{2\sigma_1^2} - \frac{(T_2-T)^2}{2\sigma_2^2}} \end{aligned} \quad (2.8)$$

Thus, given the two measurements  $T_1$  and  $T_2$  and their standard deviations  $\sigma_1$  and  $\sigma_2$ , we can find the most likely value of  $T$  by maximizing the likelihood 2.8, or even its logarithm:

$$\max_T \ln L_{\sigma_1\sigma_2}(T|T_1, T_2) = \max_T \left[ \text{constant} - \frac{(T_1 - T)^2}{2\sigma_1^2} - \frac{(T_2 - T)^2}{2\sigma_2^2} \right] \quad (2.9)$$

This maximization leads to the minimization of

$$J(T) = \frac{1}{2} \left[ -\frac{(T_1 - T)^2}{\sigma_1^2} - \frac{(T_2 - T)^2}{\sigma_2^2} \right] \quad (2.10)$$

which is the cost function 2.1 for this simplified case.

Let's consider now the more general cost function (eq. 2.1): we define the likelihood of the true state  $\mathbf{x}(t)$  given the forecast field  $\mathbf{x}^f(t)$  (used as background field) or given the new observations  $\mathbf{y}^o$  in the following way [16]:

$$L_{\mathbf{P}^f}(\mathbf{x}|\mathbf{x}^f) = p_{\mathbf{P}^f}(\mathbf{x}^f|\mathbf{x}) = \frac{1}{(2\pi)^{N/2}|\mathbf{P}^f|^{1/2}} e^{-\frac{1}{2}[(\mathbf{x}^f - \mathbf{x})^T (\mathbf{P}^f)^{-1} (\mathbf{x}^f - \mathbf{x})]} \quad (2.11)$$

$$L_{\mathbf{R}}(\mathbf{x}|\mathbf{y}^o) = p_{\mathbf{R}}(\mathbf{y}^o|\mathbf{x}) = \frac{1}{(2\pi)^{M/2}|\mathbf{R}|^{1/2}} e^{-\frac{1}{2}[(\mathbf{y}^o - H\mathbf{x})^T \mathbf{R}^{-1} (\mathbf{y}^o - H\mathbf{x})]} \quad (2.12)$$

where  $N$  is the number of components of the vectors  $\mathbf{x}(t)$  and  $\mathbf{x}^f(t)$ , while  $M$  is that of the vector  $\mathbf{y}^o$ . The joint likelihood, being independent the forecast  $\mathbf{x}^f$  and the new observations  $\mathbf{y}^o$ , is the product of the two Gaussian likelihoods:

$$L(\mathbf{x}|\mathbf{x}^f, \mathbf{y}^o) = L_{\mathbf{P}^f}(\mathbf{x}|\mathbf{x}^f) L_{\mathbf{R}}(\mathbf{x}|\mathbf{y}^o) \quad (2.13)$$

$$= \frac{e^{-\frac{1}{2}[(\mathbf{x}^f - \mathbf{x})^T (\mathbf{P}^f)^{-1} (\mathbf{x}^f - \mathbf{x})]} e^{-\frac{1}{2}[(\mathbf{y}^o - H\mathbf{x})^T \mathbf{R}^{-1} (\mathbf{y}^o - H\mathbf{x})]}}{(2\pi)^{(N+M)/2} |\mathbf{P}^f|^{1/2} |\mathbf{R}|^{1/2}} \quad (2.14)$$

The most likely analysis state  $\mathbf{x}^a$ , which maximizes the joint likelihood and its logarithm as well, also minimizes the cost function 2.1.

### 2.1.2 Bayesian approach

Let's go back to our simplified 1-dimensional case, discussed in subsection 2.1.1: there is also a Bayesian derivation for 2.10. We made the observation  $T_1$  (the forecast state in the assimilation cycle) with an *a priori* probability distribution of the truth — that is, before the second observation:

$$p_{T_1\sigma_1}(T) = \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(T_1-T)^2}{2\sigma_1^2}} \quad (2.15)$$

The Bayes Theorem for the *a posteriori* probability of the truth given the new measurement  $T_2$  is:

$$\begin{aligned} p_{\sigma_2}(T|T_2) &= \frac{p_{\sigma_2}(T_2|T) p_{T_1\sigma_1}(T)}{p_{\sigma_2}(T_2)} \\ &= \frac{\frac{1}{\sqrt{2\pi}\sigma_2} e^{-\frac{(T_2-T)^2}{2\sigma_2^2}} \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(T_1-T)^2}{2\sigma_1^2}}}{p_{\sigma_2}(T_2)} \end{aligned} \quad (2.16)$$

Since the denominator

$$p_{\sigma_2}(T_2) = \int_{T^*} \frac{1}{\sqrt{2\pi}\sigma_2} e^{-\frac{(T_2-T^*)^2}{2\sigma_2^2}} dT^*$$

is independent of  $T$ , maximizing the *a posteriori* probability 2.16 means maximizing the logarithm of the numerator, that leads again to the minimization of the cost function 2.10.

In the more general case (eq. 2.1), we suppose that the truth vector  $\mathbf{x}(t)$  is the result of a stochastic process defined by the following *a priori* probability distribution function, given the forecast field  $\mathbf{x}^f(t)$  (used as background):

$$p_{\mathbf{P}^f}(\mathbf{x}) = \frac{1}{(2\pi)^{N/2} |\mathbf{P}^f|^{1/2}} e^{-\frac{1}{2} [(\mathbf{x}^f - \mathbf{x})^T (\mathbf{P}^f)^{-1} (\mathbf{x}^f - \mathbf{x})]} \quad (2.17)$$

When we get new observations  $\mathbf{y}^o$ , the Bayes theorem gives us the *a posteriori* probability:

$$p(\mathbf{x}|\mathbf{y}^o) = \frac{p_{\mathbf{R}}(\mathbf{y}^o|\mathbf{x}) p_{\mathbf{P}^f}(\mathbf{x})}{p(\mathbf{y}^o)} \quad (2.18)$$

where  $p(\mathbf{y}^o)$  is the climatological observations distribution. Eq. 2.18 gives:

$$p(\mathbf{x}|\mathbf{y}^o) = \frac{e^{-\frac{1}{2} [(\mathbf{y}^o - H\mathbf{x})^T \mathbf{R}^{-1} (\mathbf{y}^o - H\mathbf{x}) + (\mathbf{x}^f - \mathbf{x})^T (\mathbf{P}^f)^{-1} (\mathbf{x}^f - \mathbf{x})]}}{(2\pi)^{(N+M)/2} |\mathbf{R}|^{1/2} |\mathbf{P}^f|^{1/2}} \frac{1}{p(\mathbf{y}^o)} \quad (2.19)$$

Since  $p(\mathbf{y}^o)$  does not depend on the current state  $\mathbf{x}(t)$ , the maximum of the *a posteriori* probability 2.19 will coincide with the maximum numerator, or with the minimum of the cost function 2.1.

### 2.1.3 3D-Var scheme

The analysis state that minimizes the cost function  $J(\mathbf{x})$  in eq. 2.1 is given by:

$$\nabla_{\mathbf{x}} J(\mathbf{x}^a) = 0 \quad (2.20)$$

If we assume that the analysis is a good approximation to the truth and to the observations, we can linearize the observation operator  $H$  around the background, or around the forecast if we use this as the background field:

$$\mathbf{y}^o - H(\mathbf{x}) = \mathbf{y}^o - H[\mathbf{x}^f + (\mathbf{x} - \mathbf{x}^f)] \quad (2.21)$$

$$= \mathbf{y}^o - H(\mathbf{x}^f) - \mathbf{H}(\mathbf{x} - \mathbf{x}^f) \quad (2.22)$$

After dropping the time dependence for sake of clarity, we can rearrange the expression for the cost function:

$$J(\mathbf{x}) = \frac{1}{2} [\mathbf{x} - \mathbf{x}^f]^T (\mathbf{P}^f)^{-1} [\mathbf{x} - \mathbf{x}^f] + \frac{1}{2} [H(\mathbf{x}) - \mathbf{y}^o]^T \mathbf{R}^{-1} [H(\mathbf{x}) - \mathbf{y}^o] \quad (2.23)$$

$$= \frac{1}{2} [\mathbf{x} - \mathbf{x}^f]^T (\mathbf{P}^f)^{-1} [\mathbf{x} - \mathbf{x}^f] + \frac{1}{2} [\mathbf{y}^o - H(\mathbf{x}^f) - \mathbf{H}(\mathbf{x} - \mathbf{x}^f)]^T \mathbf{R}^{-1} [\mathbf{y}^o - H(\mathbf{x}^f) - \mathbf{H}(\mathbf{x} - \mathbf{x}^f)] \quad (2.24)$$

$$= \frac{1}{2} [\mathbf{x} - \mathbf{x}^f]^T (\mathbf{P}^f)^{-1} [\mathbf{x} - \mathbf{x}^f] + \frac{1}{2} [\mathbf{x} - \mathbf{x}^f]^T \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} [\mathbf{x} - \mathbf{x}^f] + \frac{1}{2} [H(\mathbf{x}^f) - \mathbf{y}^o]^T \mathbf{R}^{-1} \mathbf{H} [\mathbf{x} - \mathbf{x}^f] + \frac{1}{2} [\mathbf{x} - \mathbf{x}^f]^T \mathbf{H}^T \mathbf{R}^{-1} [H(\mathbf{x}^f) - \mathbf{y}^o] + \frac{1}{2} [H(\mathbf{x}^f) - \mathbf{y}^o]^T \mathbf{R}^{-1} [H(\mathbf{x}^f) - \mathbf{y}^o] \quad (2.25)$$

and see that it's a quadratic function of the analysis increment  $\mathbf{x} - \mathbf{x}^f$ . Now, consider the general quadratic function

$$F(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{v}^T \mathbf{x} + k \quad (2.26)$$

where  $\mathbf{x}$  and  $\mathbf{v}$  are vectors of the same  $N$ -dimensional vector space,  $\mathbf{A}$  is an  $N \times N$  symmetric matrix and  $k$  is a scalar. It can be shown [16] that its gradient is

$$\nabla_{\mathbf{x}} F(\mathbf{x}) = \mathbf{A} \mathbf{x} + \mathbf{v} \quad (2.27)$$

So, for the cost function 2.25 the gradient with respect to  $\mathbf{x}$  is the same as that with respect to  $\mathbf{x} - \mathbf{x}^f$ :

$$\nabla J(\mathbf{x}) = \left[ (\mathbf{P}^f)^{-1} + \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} \right] (\mathbf{x} - \mathbf{x}^f) + \mathbf{H}^T \mathbf{R}^{-1} [H(\mathbf{x}^f) - \mathbf{y}^o] \quad (2.28)$$

In order to minimize it and to calculate the analysis state  $\mathbf{x}^a$ , we set  $\nabla J(\mathbf{x}^a) = 0$ , so:

$$\left[ (\mathbf{P}^f)^{-1} + \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} \right] (\mathbf{x}^a - \mathbf{x}^f) = \mathbf{H}^T \mathbf{R}^{-1} [\mathbf{y}^o - H(\mathbf{x}^f)] \quad (2.29)$$

or alternatively:

$$\mathbf{x}^a = \mathbf{x}^f + \left[ (\mathbf{P}^f)^{-1} + \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} \right]^{-1} \mathbf{H}^T \mathbf{R}^{-1} [\mathbf{y}^o - H(\mathbf{x}^f)] \quad (2.30)$$

The last equation can be expressed in term of the analysis increment  $\delta \mathbf{x}^a = \mathbf{x}^a - \mathbf{x}^f$  and the innovation  $\delta \mathbf{y}^o = \mathbf{y}^o - H(\mathbf{x}^f)$ :

$$\delta \mathbf{x}^a = \left[ (\mathbf{P}^f)^{-1} + \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} \right]^{-1} \mathbf{H}^T \mathbf{R}^{-1} \delta \mathbf{y}^o \quad (2.31)$$

which is the solution of the variational analysis problem, called 3D-Var. It can also be shown that

$$\left[ (\mathbf{P}^f)^{-1} + \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} \right]^{-1} \mathbf{H}^T \mathbf{R}^{-1} = \mathbf{P}^f \mathbf{H}^T [\mathbf{H} \mathbf{P}^f \mathbf{H}^T + \mathbf{R}]^{-1} \quad (2.32)$$

so that

$$\delta \mathbf{x}^a = \mathbf{P}^f \mathbf{H}^T [\mathbf{H} \mathbf{P}^f \mathbf{H}^T + \mathbf{R}]^{-1} \delta \mathbf{y}^o \quad (2.33)$$

or equivalently:

$$\mathbf{x}^a = \mathbf{x}^f + \mathbf{P}^f \mathbf{H}^T [\mathbf{H} \mathbf{P}^f \mathbf{H}^T + \mathbf{R}]^{-1} [\mathbf{y}^o - H(\mathbf{x}^f)] \quad (2.34)$$

Subtracting the true state vector  $\mathbf{x}^t$  in both lhs and rhs, we get an expression for the analysis error:

$$\boldsymbol{\eta}^a = \boldsymbol{\eta}^f + \mathbf{P}^f \mathbf{H}^T [\mathbf{H} \mathbf{P}^f \mathbf{H}^T + \mathbf{R}]^{-1} [\mathbf{y}^o - H(\mathbf{x}^f)] \quad (2.35)$$

If we define the *gain matrix*

$$\mathbf{K} = \mathbf{P}^f \mathbf{H}^T [\mathbf{H} \mathbf{P}^f \mathbf{H}^T + \mathbf{R}]^{-1} \quad (2.36)$$

the equations 2.33 and 2.34 become:

$$\delta \mathbf{x}^a = \mathbf{K} \delta \mathbf{y}^o \quad (2.37)$$

$$\mathbf{x}^a = [\mathbf{I} - \mathbf{K} \mathbf{H}] \mathbf{x}^f + \mathbf{K} \mathbf{y}^o \quad (2.38)$$

If the forecast error  $\boldsymbol{\eta}^f$  is small, we can linearize the observation operator  $H$  and the equation  $H(\mathbf{x} + \delta\mathbf{x}) = H(\mathbf{x}) + \mathbf{H}\delta\mathbf{x}$  holds true; so equation 2.35 yields:

$$\boldsymbol{\eta}^a = \boldsymbol{\eta}^f + \mathbf{K}\mathbf{y}^o - \mathbf{K}H(\mathbf{x}^f) \quad (2.39)$$

$$= \boldsymbol{\eta}^f + \mathbf{K}\mathbf{y}^o - \mathbf{K}H(\mathbf{x}^t + \mathbf{x}^f - \mathbf{x}^t) \quad (2.40)$$

$$= \boldsymbol{\eta}^f + \mathbf{K}\mathbf{y}^o - \mathbf{K}H(\mathbf{x}^t + \boldsymbol{\eta}^f) \quad (2.41)$$

$$= \boldsymbol{\eta}^f + \mathbf{K}\mathbf{y}^o - \mathbf{K}H(\mathbf{x}^t) - \mathbf{K}H(\boldsymbol{\eta}^f) \quad (2.42)$$

$$= [\mathbf{I} - \mathbf{K}H]\boldsymbol{\eta}^f + \mathbf{K}\delta\mathbf{y}^o \quad (2.43)$$

Since we may assume that the forecast error and the observation error are uncorrelated, the analysis covariance matrix may be written as follows:

$$\mathbf{P}^a = \langle \boldsymbol{\eta}^a (\boldsymbol{\eta}^a)^T \rangle \quad (2.44)$$

$$= [\mathbf{I} - \mathbf{K}H]\mathbf{P}^f[\mathbf{I} - \mathbf{K}H]^T + \mathbf{K}\mathbf{R}\mathbf{K}^T \quad (2.45)$$

$$= [\mathbf{I} - \mathbf{K}H]\mathbf{P}^f \quad (2.46)$$

where the last equation has been derived by inserting into eq. 2.45 the expression for the gain matrix  $\mathbf{K} = \mathbf{P}^f\mathbf{H}^T[\mathbf{H}\mathbf{P}^f\mathbf{H}^T + \mathbf{R}]^{-1}$ . Indeed, from eq. 2.45 we have:

$$\mathbf{P}^a = [\mathbf{I} - \mathbf{K}H]\mathbf{P}^f - [\mathbf{I} - \mathbf{K}H]\mathbf{P}^f\mathbf{H}^T\mathbf{K}^T + \mathbf{K}\mathbf{R}\mathbf{K}^T \quad (2.47)$$

$$= [\mathbf{I} - \mathbf{K}H]\mathbf{P}^f - [\mathbf{P}^f\mathbf{H}^T - \mathbf{K}\mathbf{H}\mathbf{P}^f\mathbf{H}^T]\mathbf{K}^T + \mathbf{K}\mathbf{R}\mathbf{K}^T \quad (2.48)$$

Now, the two last terms in eq. 2.48 actually vanish, because

$$\mathbf{K} = \mathbf{P}^f\mathbf{H}^T[\mathbf{H}\mathbf{P}^f\mathbf{H}^T + \mathbf{R}]^{-1} \iff \mathbf{K}\mathbf{R} = \mathbf{P}^f\mathbf{H}^T - \mathbf{K}\mathbf{H}\mathbf{P}^f\mathbf{H}^T \quad (2.49)$$

## 2.2 Sequential assimilation

The sequential assimilation methods have a probabilistic approach to estimate the state of a system. The basic idea is to project information ahead in time and to assimilate observational data when available. We don't need to compute the adjoint model: sequential assimilation schemes are suitable for different models.

In the following subsections we will talk about Kalman Filter for linear systems and Extended Kalman Filter for nonlinear ones. In the end, we will drop a note about the very promising Ensemble Kalman Filter.

### 2.2.1 Kalman Filter

The Kalman Filter, hereafter referred to as KF, was first formulated by Kalman (1960) and Kalman and Bucy (1961), so sometimes it is called Kalman-Bucy Filter. The KF deals with linear stochastic dynamical systems where noisy observations are taken at discrete times. It is an *optimal recursive data processing algorithm*, where 'optimal' refers to the fact that it uses all available information we can provide: it processes all measurements, when available, regardless of their precision, by using [23]:

- the knowledge of the dynamics of the system and measurement device dynamics
- the statistical description of the system noises, measurement errors and model error
- any available information about initial conditions of the system
- it does *not* need old data to be kept in storage

The KF is basically a set of equations that implements a prediction-correction estimator: if some conditions are met, the estimator minimizes the estimated error covariance [23]. If the conditions are not fully satisfied, often the KF still works quite well. A sketch description of how it works is shown in Fig. 2.1: the prediction stage consists of the first two equations, which basically project ahead the analysis state and the analysis error covariance matrix, providing the forecast vector and the forecast error covariance matrix. The correction stage consists of further three equations, and takes advantage of new observations to provide the new analysis vector and the analysis error covariance matrix. Then, recursively, a new projection ahead is performed till new observations become available.

Let's consider a linear discrete stochastic dynamical system of the form of the eq. 1.11, i.e.

$$\mathbf{x}_{k+1}^t = \mathbf{M}\mathbf{x}_k^t + \mathbf{q}_{k+1} \quad (2.50)$$

Here the linear model operator  $\mathbf{M}$  projects information ahead from time  $t_k$  to time  $t_{k+1}$ , so actually

$$\mathbf{M} = \mathbf{M}(t_k, t_{k+1}) \quad (2.51)$$

We will use this simplified notation in equations below. The term  $\mathbf{q}_{k+1}$  in eq. 2.50 is a white Gaussian noise with zero mean and covariance matrix  $\mathbf{Q}_{k+1}$ , namely

$$\langle \mathbf{q}_{k+1} \rangle = 0 \quad (2.52)$$

$$\mathbf{Q}_{k+1} = \langle \mathbf{q}_{k+1} \mathbf{q}_{k+1}^T \rangle \quad (2.53)$$

and may represent subgrid-scale processes not resolved by the model [14]. Under these conditions the KF is described by the following set of equations:

$$\mathbf{x}_{k+1}^f = \mathbf{M}\mathbf{x}_k^a \quad (2.54)$$

$$\mathbf{P}_{k+1}^f = \mathbf{M}\mathbf{P}_k^a\mathbf{M}^T + \mathbf{Q}_{k+1} \quad (2.55)$$

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1}^f\mathbf{H}^T(\mathbf{H}\mathbf{P}_{k+1}^f\mathbf{H}^T + \mathbf{R})^{-1} \quad (2.56)$$

$$\mathbf{x}_{k+1}^a = (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H})\mathbf{x}_{k+1}^f + \mathbf{K}_{k+1}\mathbf{y}_{k+1}^o \quad (2.57)$$

$$\mathbf{P}_{k+1}^a = (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H})\mathbf{P}_{k+1}^f \quad (2.58)$$

where, following usual conventions:

|                |  |
|----------------|--|
| $\mathbf{x}^t$ | is the true state  |
| $\mathbf{x}^a$ | is the analysis state  |
| $\mathbf{x}^f$ | the forecast state   |
| $\mathbf{y}^o$ | the observation vector   |
| $\mathbf{P}^a$ | the analysis error covariance matrix   |
| $\mathbf{P}^f$ | the forecast error covariance matrix   |
| $\mathbf{R}$   | the observational error covariance matrix  |
| $H$            | the (possibly nonlinear) observation operator  |
| $\mathbf{H}$   | the linearized observation operator (it transforms vectors in model space to vectors in observation space) |
| $\mathbf{M}$   | the linear model operator  |
| $\mathbf{Q}$   | the forecast model error covariance matrix   |
| $\mathbf{K}$   | the gain matrix  |
| $\mathbf{I}$   | the identity matrix.   |

Subscripts in the KF equations indicate the time steps where new observations are available.

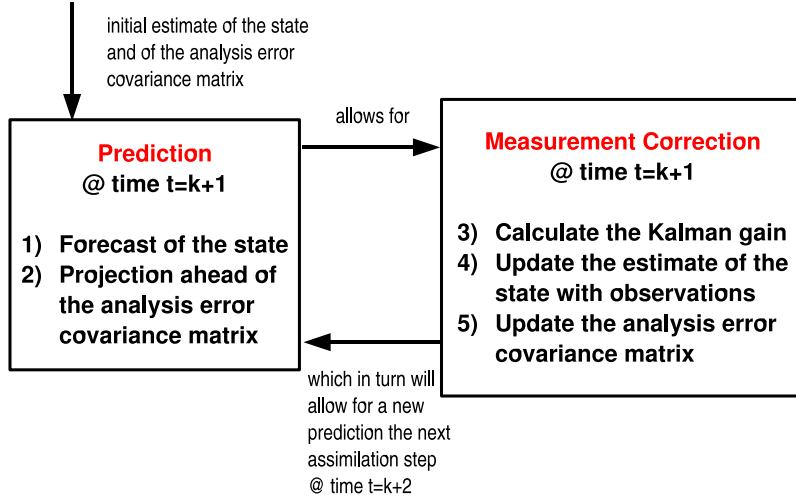
If we compute the expectation value of eq. 2.50 we get:

$$\langle \mathbf{x}_{k+1}^t \rangle = \langle \mathbf{M}\mathbf{x}_k^t + \mathbf{q}_{k+1} \rangle \quad (2.59)$$

$$= \langle \mathbf{M}\mathbf{x}_k^t \rangle \quad (2.60)$$

$$= \mathbf{M} \langle \mathbf{x}_k^t \rangle \quad (2.61)$$

Figure 2.1: How the Kalman Filter works.



or, in the usual data assimilation notation:

$$\mathbf{x}_{k+1}^f = \mathbf{M}\mathbf{x}_k^a \quad (2.62)$$

which is eq. 2.54. It should be noticed that, due to linearity of this special case, if the initial PDF is Gaussian, so it will remain for any future time: we can have a complete description of the future PDF by its mean and covariance. So we need the covariance matrix  $\mathbf{P}^f$  evolution, from time  $t_k$  to time  $t_{k+1}$ . It can be derived by rearranging the forecast error definition, together with the linearity of the model operator  $\mathbf{M}$ , and by using equations 2.50, 2.62:

$$\boldsymbol{\eta}_{k+1}^f = \mathbf{x}_{k+1}^f - \mathbf{x}_{k+1}^t \quad (2.63)$$

$$= \mathbf{M}\mathbf{x}_k^a - \mathbf{M}\mathbf{x}_k^t - \mathbf{q}_{k+1} \quad (2.64)$$

$$= \mathbf{M}(\mathbf{x}_k^a - \mathbf{x}_k^t) - \mathbf{q}_{k+1} \quad (2.65)$$

$$= \mathbf{M}\boldsymbol{\eta}_k^a - \mathbf{q}_{k+1} \quad (2.66)$$

Since  $\mathbf{q}_{k+1}$  is a white Gaussian noise with zero mean, the expectation value of  $\boldsymbol{\eta}_{k+1}^f$  reads:

$$\langle \boldsymbol{\eta}_{k+1}^f \rangle = \mathbf{M} \langle \boldsymbol{\eta}_k^a \rangle \quad (2.67)$$



Now, if the analysis error and the model error are uncorrelated, multiplying  $\boldsymbol{\eta}_{k+1}^f$  by its transpose and taking the expectation value, we get:

$$\mathbf{P}_{k+1}^f = \langle \boldsymbol{\eta}_{k+1}^f (\boldsymbol{\eta}_{k+1}^f)^T \rangle \quad (2.68)$$

$$= \langle (\mathbf{M}\boldsymbol{\eta}_k^a - \mathbf{q}_{k+1})(\mathbf{M}\boldsymbol{\eta}_k^a - \mathbf{q}_{k+1})^T \rangle \quad (2.69)$$

$$= \mathbf{M} \langle \boldsymbol{\eta}_k^a (\boldsymbol{\eta}_k^a)^T \rangle \mathbf{M}^T + \langle \mathbf{q}_{k+1} (\mathbf{q}_{k+1})^T \rangle \quad (2.70)$$

$$= \mathbf{M} \mathbf{P}_k^a \mathbf{M}^T + \mathbf{Q}_{k+1} \quad (2.71)$$

which is eq. 2.55.

The analysis state is computed as in 3D-Var assimilation scheme, eq. 2.34

$$\mathbf{x}^a = \mathbf{x}^f + \mathbf{K}_{k+1} [\mathbf{y}^o - H(\mathbf{x}^f)] \quad (2.72)$$

or even

$$\mathbf{x}^a = [\mathbf{I} - \mathbf{K}_{k+1} H] \mathbf{x}^f + \mathbf{K}_{k+1} \mathbf{y}^o \quad (2.73)$$

where the Kalman gain  $\mathbf{K}_{k+1}$  is defined by

$$\mathbf{K}_{k+1} = \mathbf{P}^f \mathbf{H}^T [\mathbf{H} \mathbf{P}^f \mathbf{H}^T + \mathbf{R}]^{-1} \quad (2.74)$$

The analysis error covariance matrix is given by eq. 2.46, i.e.:

$$\mathbf{P}^a = (\mathbf{I} - \mathbf{K}_{k+1} \mathbf{H}) \mathbf{P}_{k+1}^f \quad (2.75)$$

## 2.2.2 Extended Kalman Filter

One of the methods devised to address the problem of estimating the initial conditions for a forecast model is the *Extended Kalman Filter* (EKF), where the term 'extended' refers to the Kalman Filter's approximation for nonlinear systems.

The EKF is described by the following set of equations (see for example [14]):

$$\mathbf{x}_{k+1}^f = \mathbf{M} \mathbf{x}_k^a \quad (2.76)$$

$$\mathbf{P}_{k+1}^f = \mathbf{M} \mathbf{P}_k^a \mathbf{M}^T + \mathbf{Q}_k \quad (2.77)$$

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1}^f \mathbf{H}^T (\mathbf{H} \mathbf{P}_{k+1}^f \mathbf{H}^T + \mathbf{R})^{-1} \quad (2.78)$$

$$\mathbf{x}_{k+1}^a = (\mathbf{I} - \mathbf{K}_{k+1} H) \mathbf{x}_{k+1}^f + \mathbf{K}_{k+1} \mathbf{y}_{k+1}^o \quad (2.79)$$

$$\mathbf{P}_{k+1}^a = (\mathbf{I} - \mathbf{K}_{k+1} \mathbf{H}) \mathbf{P}_{k+1}^f \quad (2.80)$$

where we used the common conventions as for the KF, and the *nonlinear* model operator  $M$ . Here  $\mathbf{M}$  is no longer the linear model operator, but the Tangent Linear Model operator. Subscripts indicate the time steps where new observations are available and, again, we dropped them for operators  $M$ ,  $\mathbf{M}$ ,  $H$ ,  $\mathbf{H}$ .

For EKF, due to the lack of linearity of the model operator  $M$ , equation 2.66 for the forecast error is no longer valid; nonetheless it may be rewritten in an approximate form, by using the tangent linear model operator and equation  $\mathbf{x}_{k+1}^t = M\mathbf{x}_k^t + \mathbf{q}_{k+1}$ :

$$\boldsymbol{\eta}_{k+1}^f = \mathbf{x}_{k+1}^f - \mathbf{x}_{k+1}^t \quad (2.81)$$

$$= M\mathbf{x}_k^a - M\mathbf{x}_k^t - \mathbf{q}_{k+1} \quad (2.82)$$

$$\simeq \mathbf{M}(\mathbf{x}_k^a - \mathbf{x}_k^t) - \mathbf{q}_{k+1} \quad (2.83)$$

So, since

$$\boldsymbol{\eta}_{k+1}^f \simeq \mathbf{M}\boldsymbol{\eta}_k^a - \mathbf{q}_{k+1} \quad (2.84)$$

is only an approximate expression, the corresponding expression for the forecast error covariance matrix (eq. 2.71) will be approximate as well:

$$\mathbf{P}_{k+1}^f \simeq \mathbf{M}\mathbf{P}_k^a\mathbf{M}^T + \mathbf{Q}_k \quad (2.85)$$

The EKF, after an initial transient, should give *both* the best linear unbiased estimate of the state of the system *and* its error covariance. But if the system is (locally) highly nonlinear, or should the observations be not adequately frequent, the linearization may become a hypothesis which is not actually fulfilled: that can jeopardize the stability of the filter and the filter may diverge [16]. Furthermore, for realistic NWP model the EKF can *not* be implemented due to both the prohibitive computational costs in estimating the covariance matrices and the uncertainties about the model error.

### 2.2.3 Ensemble Kalman Filter

In the *Ensemble Kalman Filter* (EnKF) approach, proposed by Evensen in 1994 [10], an ensemble of  $N_{ens}$  data assimilation cycles are run simultaneously and independently: all of them assimilate the same set of observations, but for each member of the ensemble it will be added a different random perturbation to the observations. This ensemble can be used to estimate the forecast error covariance matrix  $\mathbf{P}^f$ : after computing the analysis state  $\mathbf{x}_j^a(t_k)$ ,  $j \in \{1, \dots, N_{ens}\}$ ,

for each member of the ensemble, we can obtain the forecast states:

$$\mathbf{x}_j^f(t_{k+1}) = M_k^j \mathbf{x}_j^a(t_k) \quad j \in \{1, \dots, N_{ens}\} \quad (2.86)$$

the ensemble average  $\langle \mathbf{x}^f(t_{k+1}) \rangle$  and an estimate of the forecast error covariance matrix, a sort of average of the  $N_{ens}$  forecast error covariance matrices, e.g.:

$$\mathbf{P}_{k+1}^f \simeq \frac{1}{N_{ens} - 1} \sum_{j=1}^{N_{ens}} \left[ \mathbf{x}_j^f(t_{k+1}) - \langle \mathbf{x}^f(t_{k+1}) \rangle \right] \left[ \mathbf{x}_j^f(t_{k+1}) - \langle \mathbf{x}^f(t_{k+1}) \rangle \right]^T \quad (2.87)$$

This will actually tend to underestimate the forecast error covariance matrix  $\mathbf{P}^f$ : other estimate can be devised [16].

The EnKF approach has many advantages, among which:

- since typically  $N_{ens}$  is somewhere between 10 and 100, the computational cost of EnKF is increased by the same factor with respect to 3D-Var, for example. But it's much smaller compared to that of an EKF
- EnKF does not need a linear or adjoint model
- it does not even require the linearization of the evolution of the forecast error covariance matrix  $\mathbf{P}^f$

Despite EnKF is not yet implemented in operational NWP forecast, it seems nowadays one of the most promising assimilation schemes for the future.

## 2.3 AUS: Assimilation in the Unstable Subspace

The *Assimilation in the Unstable Subspace* (AUS) was introduced by Trevisan and Ubaldi in 2004 [33], hereafter referred to as TU, and developed by Trevisan, Ubaldi and Carrassi [34, 35, 3], to minimize the analysis and forecast errors by exploiting the flow-dependent instabilities of the forecast-analysis cycle system, which may be thought of as a system forced by observations. In the AUS scheme the assimilation is obtained by confining the analysis increment  $\delta \mathbf{x}^a = \mathbf{x}^a - \mathbf{x}^f$  in the unstable subspace of the forecast-analysis cycle system so that it will have the same structure of the dominant instabilities of the system. In such a way the dynamically unstable components, present in the forecast error, which are responsible for error growth, are in principle systematically reduced or eliminated. The unstable subspace will be estimated by *breeding on the data assimilation system* (BDAS), a technique to be discussed below. TU showed that AUS is a reliable and efficient approach in the 40 variables Lorenz 1996

model [20], while the subsequent studies proved the same for different, more realistic models and observational configurations, including a Quasi-Geostrophic model with 14784 degrees of freedom [3], and a high dimensional, primitive equation ocean model with 301120 degrees of freedom [35]; the experiments encompassed fixed and “adaptive”, or “targeted”, observations. In these contexts, the AUS-BDAS dynamical system approach greatly reduces the analysis error, with reasonable computational costs for data assimilation with respect, for example, to a prohibitive full Extended Kalman Filter approach. This is a follow-up study in which we will revisit the AUS-BDAS approach in the more basic, highly nonlinear Lorenz 1963 convective model.

In practice, in the same spirit as of the EKF approach (equations 2.76-2.80), the AUS assimilation algorithm is aimed at finding a simplified form of the forecast covariance matrix  $\mathbf{P}^f$  by exploiting the local unstable structures of the forecast-analysis cycle system, which in turn are estimated by BDAS. Once  $\mathbf{P}^f$  has been estimated, an approximated gain matrix  $\mathbf{K}$  may be computed, so finally — with some knowledge of the observational error covariance matrix  $\mathbf{R}$  and the observation operator  $H$  — we can estimate the analysis vector  $\mathbf{x}^a$ .

### 2.3.1 AUS: how it works

The forecast error is regarded as made of two components, the first on the unstable subspace, and the other one on the complementary subspace [35]:

$$\boldsymbol{\eta}^f = \mathbf{E}\boldsymbol{\gamma} + \boldsymbol{\xi} \quad (2.88)$$

where the matrix  $\mathbf{E}$  stores in its columns the normalized estimated unstable directions, the column vector  $\boldsymbol{\gamma}$  represents the forecast error component in the unstable basis: so  $\mathbf{E}\boldsymbol{\gamma}$  is the linear combination of the unstable directions that represents the forecast error component on the unstable subspace. The correspondent forecast covariance matrix may be derived:

$$\mathbf{P}^f = \langle \boldsymbol{\eta}^f (\boldsymbol{\eta}^f)^T \rangle \quad (2.89)$$

$$= \mathbf{E} \langle \boldsymbol{\gamma} \boldsymbol{\gamma}^T \rangle \mathbf{E}^T + \mathbf{E} \langle \boldsymbol{\gamma} \boldsymbol{\xi}^T \rangle + \langle \boldsymbol{\xi} \boldsymbol{\gamma}^T \rangle \mathbf{E}^T + \langle \boldsymbol{\xi} \boldsymbol{\xi}^T \rangle \quad (2.90)$$

If we set  $\boldsymbol{\Gamma} = \langle \boldsymbol{\gamma} \boldsymbol{\gamma}^T \rangle$  and assume that the forecast error component in the complementary subspace is small, we can neglect in eq. 2.90 all term containing  $\boldsymbol{\xi}$ , and the  $\mathbf{P}^f$  may be approximated:

$$\mathbf{P}^f \simeq \mathbf{E} \boldsymbol{\Gamma} \mathbf{E}^T \quad (2.91)$$

The corresponding gain matrix is approximated as well:

$$\mathbf{K} = \mathbf{P}^f \mathbf{H}^T (\mathbf{H} \mathbf{P}^f \mathbf{H}^T + \mathbf{R})^{-1} \quad (2.92)$$

$$\simeq \mathbf{E} \mathbf{\Gamma} \mathbf{E}^T \mathbf{H}^T (\mathbf{H} \mathbf{E} \mathbf{\Gamma} \mathbf{E}^T \mathbf{H}^T + \mathbf{R})^{-1} \quad (2.93)$$

where  $\mathbf{R}$  is the usual observational error covariance matrix and  $\mathbf{H}$  is the Jacobian of the possibly nonlinear observation operator  $H$ . The analysis vector expression reads:

$$\mathbf{x}^a = \mathbf{x}^f + \mathbf{K} [\mathbf{y}^o - H(\mathbf{x}^f)] \quad (2.94)$$

$$\simeq \mathbf{x}^f + \mathbf{E} \mathbf{\Gamma} \mathbf{E}^T \mathbf{H}^T (\mathbf{H} \mathbf{E} \mathbf{\Gamma} \mathbf{E}^T \mathbf{H}^T + \mathbf{R})^{-1} [\mathbf{y}^o - H(\mathbf{x}^f)] \quad (2.95)$$

The approximated equation 2.91, written by neglecting the component of  $\mathbf{P}^f$  out of the unstable subspace spanned by the basis vector stored in the matrix  $\mathbf{E}$ , results in a gain matrix  $\mathbf{K}$  computed in a subspace (eq. 2.93). The resulting analysis of eq. 2.95 reduces the error component in such subspace [35].

If  $\mathbf{M}$  is the tangent linear propagator (between time  $t_k$  to time  $t_{k+1}$ , [3]) we can write down:

$$\mathbf{M} \mathbf{E}_k = \mathbf{E}_{k+1} \mathbf{\Lambda}_k$$

where  $\mathbf{\Lambda}_k$  is the matrix whose diagonal elements are the amplification factors,  $\exp \left( \int_{t_k}^{t_{k+1}} \lambda_i(t) dt \right)$ , where  $\lambda_i$  are the local Lyapunov exponents, in decreasing order, corresponding to the  $i$ -th column vector of  $\mathbf{E}_k$ . The forecast error will then evolve according to:

$$\boldsymbol{\eta}_{k+1}^f = \mathbf{E}_{k+1} \mathbf{\Lambda}_k \mathbf{E}_k^{-1} \boldsymbol{\eta}_k^a \quad (2.96)$$

### 2.3.2 AUS: a simple example

For example, if the state vector has dimension 3, the number of observations each assimilation step is 2 and the number of unstable directions is 2, the  $3 \times 2$  matrix  $\mathbf{E}$  is:

$$\mathbf{E} = \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \\ e_{31} & e_{32} \end{bmatrix} \quad (2.97)$$

where  $\mathbf{e}_1 = [e_{11} \ e_{21} \ e_{31}]^T$  and  $\mathbf{e}_2 = [e_{12} \ e_{22} \ e_{32}]^T$  are the 2 unstable directions. The  $2 \times 1$  column vector  $\boldsymbol{\gamma}$  is

$$\boldsymbol{\gamma} = \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix} \quad (2.98)$$

so the  $3 \times 1$  product  $\mathbf{E}\boldsymbol{\gamma}$  is

$$\mathbf{E}\boldsymbol{\gamma} = \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \\ e_{31} & e_{32} \end{bmatrix} \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix} \quad (2.99)$$

$$= \begin{bmatrix} e_{11}\gamma_1 + e_{12}\gamma_2 \\ e_{21}\gamma_1 + e_{22}\gamma_2 \\ e_{31}\gamma_1 + e_{32}\gamma_2 \end{bmatrix} \quad (2.100)$$

The forecast error covariance matrix is  $3 \times 3$ :

$$\mathbf{P}^f \simeq \mathbf{E}\boldsymbol{\Gamma}\mathbf{E}^T \quad (2.101)$$

$$\simeq \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \\ e_{31} & e_{32} \end{bmatrix} \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix} [\gamma_1 \ \gamma_2] \begin{bmatrix} e_{11} & e_{21} & e_{31} \\ e_{12} & e_{22} & e_{32} \end{bmatrix} \quad (2.102)$$

where  $\boldsymbol{\Gamma}$  is the  $2 \times 2$  matrix

$$\boldsymbol{\Gamma} = \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix} [\gamma_1 \ \gamma_2] \quad (2.103)$$

$$= \begin{bmatrix} \gamma_1\gamma_1 & \gamma_1\gamma_2 \\ \gamma_1\gamma_2 & \gamma_2\gamma_2 \end{bmatrix} \quad (2.104)$$

As we already mentioned in subsection 1.6.5 for a similar case, the observation error covariance  $\mathbf{R}$  is  $2 \times 2$ , the observation operator  $\mathbf{H}$  is  $2 \times 3$ , and the gain matrix  $\mathbf{K}$  is  $3 \times 2$ .

### 2.3.3 Refresh procedure

After the analysis, part of the information in the bred vectors, used to estimate the unstable structures of the system, will be no longer available: that's why a “refresh” procedure may improve our capabilities to capture the system's instabilities. “Refresh” means that a new random perturbation is introduced in the place of bred vectors already used in assimilation; which in turn may be simply discarded or “recycled” by adding them to the other vectors.

Which is the better strategy depends on the complexity of our system, and an “in between” approach may be also implemented [35].

### 2.3.4 Using a single bred vector for assimilation

When we use a single bred vector to estimate the unstable 1-dimensional subspace, the matrix  $\mathbf{E}$  reduces to a single column vector  $\mathbf{e}$ , the matrix  $\mathbf{\Gamma}$  becomes the scalar  $\gamma^2$  and the expressions discussed in subsection 2.3.1 for forecast error, forecast error covariance matrix and gain matrix reduce to:

$$\boldsymbol{\eta}^f = \mathbf{e}\gamma + \boldsymbol{\xi} \quad (2.105)$$

$$\mathbf{P}^f \simeq \gamma^2 \mathbf{e}\mathbf{e}^T \quad (2.106)$$

$$\mathbf{K} = \gamma^2 \mathbf{e}\mathbf{e}^T \mathbf{H}^T (\gamma^2 \mathbf{H}\mathbf{e}\mathbf{e}^T \mathbf{H}^T + \mathbf{R})^{-1} \quad (2.107)$$

### 2.3.5 Adaptive observation strategy

The basic idea underlying adaptive, or targeted, observations is to take measurements where the unstable structures have the maximum amplitude. The same structures exploited to locate adaptive observations are used to estimate the  $\mathbf{P}^f$ ,  $\mathbf{K}$  and analysis state  $\mathbf{x}^a$ , through equations 2.91, 2.93 and 2.95 respectively. This approach has been already tested in previous works by Trevisan, Uboldi and Carrassi and proved to be highly efficient in realistic contexts [33, 34, 3].

## 2.4 BDAS: Breeding on the Data Assimilation System

*Breeding on the Data Assimilation System* (BDAS) is a method devised to estimate the unstable structures of the data assimilation system, that can be thought of as a system forced by observations. It is a modified formulation of *breeding*: the basic idea of BDAS is to breed initially random perturbations of the analysis and to impose them the same dynamics as the analysis-forecast solution, including assimilation of the observations whenever available. The perturbations need to be evolved for a sufficiently long time for a reliable estimate of the instabilities: the corresponding time is the *breeding time*.

### 2.4.1 Standard breeding method

The breeding method is a nonlinear, finite-amplitude generalization of the algorithm used to compute the leading Lyapunov vector: as we already mentioned in subsection 1.3.9, bred vectors are indeed closely related to Local Lyapunov Vectors. This is the basic idea: A small pertur-

bation of the state, if its amplitude is periodically scaled down to be kept small, will evolve in a linear combination of unstable directions [35]. These directions are estimated by integrating the nonlinear model and by using one or more perturbed states. In realistic geophysical models the normalization amplitude, the length of the breeding time and the frequency of the normalization procedure may be tuned to filter out unwanted instabilities, such as convection [16].

In practice, standard breeding works as follows. Given a dynamical system in the form of a flow, a breeding cycle is started by adding a random initial perturbation with a fixed initial amplitude, which is introduced only once, at the beginning of the procedure. The same nonlinear model is integrated from the unperturbed (“control”) and from the perturbed initial conditions. At regular time intervals the control forecast is subtracted from the perturbed forecast, and the resulting difference is scaled down to the initial amplitude. Then it is added to the corresponding new analysis or model state [16]. The forecast state can be computed by

$$\mathbf{x}_{k+1}^f = M \mathbf{x}_k^a \quad (2.108)$$

and the perturbation dynamics will be described by

$$\delta \mathbf{x}_{k+1}^f = \mathbf{M} \delta \mathbf{x}_k^a \quad (2.109)$$

where  $M$  is the model and  $\mathbf{M} = \mathbf{M}(\mathbf{x}(t))$  its Jacobian evaluated around the forecast trajectory  $\mathbf{x}(t)$ ,  $\forall t \in ]t_k, t_{k+1}]$ .

### 2.4.2 BDAS: how it works

Since BDAS is a particular implementation of the breeding method, it shares with it the basic principles, i.e. the evolution, by using the nonlinear model, of a random initial perturbation added to the control trajectory. In the analysis step we have observational data to assimilate, and the evolved random initial perturbation will undergo the assimilation procedure.

Whenever observations become available, the analysis state is computed by

$$\mathbf{x}_{k+1}^a = [\mathbf{I} - \mathbf{K}H] \mathbf{x}_{k+1}^f + \mathbf{K} \mathbf{y}_{k+1}^o \quad (2.110)$$

or even, using eq. 2.108:

$$\mathbf{x}_{k+1}^a = [\mathbf{I} - \mathbf{K}H] M \mathbf{x}_k^a + \mathbf{K} \mathbf{y}_{k+1}^o \quad (2.111)$$

where  $\mathbf{K}$  is the gain matrix and  $H$  is the (possibly nonlinear) observation operator. The



expression 2.110 is the same in Kalman Filters and 3D-Var schemes. The perturbation equation for the system undergoing observational forcing reads:

$$\delta \mathbf{x}_{k+1}^a = [\mathbf{I} - \mathbf{K}\mathbf{H}] \delta \mathbf{x}_{k+1}^f \quad (2.112)$$

$$= [\mathbf{I} - \mathbf{K}\mathbf{H}] \mathbf{M} \delta \mathbf{x}_k^a \quad (2.113)$$

where we used the linearized observation operator  $\mathbf{H}$ , i.e. the Jacobian of the observation operator  $H$ . Breeding on the Data Assimilation System is based, rather than on eq. 2.109, on eq. 2.113, in which the matrix operator  $[\mathbf{I} - \mathbf{K}\mathbf{H}]$  has a general stabilizing effect: the assimilation will reduce the amplifying components of the error.

So, basically: in an assimilation system where observations are available once in a while, during the forecast time the free system instabilities dominate the error growth; in the analysis step the assimilation of observations will in general reduce some fast-growing components of the error.

Another important issue is the *breeding time*, i.e. the time needed for the perturbations to capture the most unstable structures. It can not be infinite, but should be long enough to provide a meaningful (set of) bred vector(s). Typically, the breeding time  $\Delta t$  is longer than the assimilation window  $\tau$ , and often is set as a multiple of  $\tau$ :

$$\Delta t = n\tau \quad (2.114)$$

### 2.4.3 BDAS, an example of practical implementation

Just to focus on a specific example, let's suppose that we deal with a simple low dimensional system for which the breeding time  $\Delta t = 2\tau$ , where  $\tau = t_{k+1} - t_k$  is the fixed assimilation window. We assume that:

- the unstable subspace is estimated by using a single bred vector
- we can discard bred vector after use

If  $t_k$  is the previous assimilation step, we use the general equation 1.34, that is:

$$\delta \mathbf{x}(t) = \frac{\partial \mathbf{M}}{\partial \mathbf{x}} \delta \mathbf{x}(t_k) = \mathbf{M} \delta \mathbf{x}(t_k), \quad t \in [t_k, t_{k+1}] \quad (2.115)$$

to estimate the evolution of the perturbation  $\delta \mathbf{x}(t)$ , for  $t$  spanning from the previous assimilation step  $t_k$  up to the new one  $t_{k+1}$ . At new assimilation step  $t_{k+1}$  we will use the evolved perturbation  $\delta \mathbf{x}(t_{k+1})$  to estimate the unstable direction of the system, after which it is dis-

Table 2.1: Breeding on the Data Assimilation System: introducing the perturbations and estimating the unstable subspace. This is a specific example in which the breeding time is  $\Delta t = 2\tau$ , where  $\tau = t_{k+1} - t_k$  is the assimilation window.

| Time                       | introduced                  | evolving  | used to assimilate<br>and then discarded | undergoing<br>assimilation          |
|----------------------------|-----------------------------|---|--|-------------------------------------|
| $t_{k-1}$                  | $\delta\mathbf{x}(t_{k-1})$ | —   | evolved $\delta\mathbf{x}(t_{k-3})$      | evolved $\delta\mathbf{x}(t_{k-2})$ |
| $t_{k-1} < t \leq t_k$     | —                           | $\delta\mathbf{x}(t_{k-2})$ & $\delta\mathbf{x}(t_{k-1})$ | —  | —                                   |
| $t_k$                      | $\delta\mathbf{x}(t_k)$     | —   | evolved $\delta\mathbf{x}(t_{k-2})$      | evolved $\delta\mathbf{x}(t_{k-1})$ |
| $t_k < t \leq t_{k+1}$     | —                           | $\delta\mathbf{x}(t_{k-1})$ & $\delta\mathbf{x}(t_k)$     | —  | —                                   |
| $t_{k+1}$                  | $\delta\mathbf{x}(t_{k+1})$ | —   | evolved $\delta\mathbf{x}(t_{k-1})$      | evolved $\delta\mathbf{x}(t_k)$     |
| $t_{k+1} < t \leq t_{k+2}$ | —                           | $\delta\mathbf{x}(t_k)$ & $\delta\mathbf{x}(t_{k+1})$     | —  | —                                   |
| $t_{k+2}$                  | $\delta\mathbf{x}(t_{k+2})$ | —   | evolved $\delta\mathbf{x}(t_k)$          | evolved $\delta\mathbf{x}(t_{k+1})$ |

carded. The already grown perturbation  $\delta\mathbf{x}(t_k)$  will undergo the same assimilation process used to evaluate the new analysis state. Furthermore, a new random perturbation  $\delta\mathbf{x}(t_{k+1})$  is introduced (refresh, see subsection 2.3.3), to be used at assimilation step  $t_{k+3}$ . Let's summarize:

In the particular example at hand, with a breeding time  $\Delta t = 2\tau$ , we will recursively follow this procedure (see Table 2.1):

1. At each new assimilation step  $t_{k+1}$ :
  - (a) We exploit the evolved perturbation  $\delta\mathbf{x}(t_{k-1})$ , previously introduced at time  $t_{k-1}$  and assimilated at time  $t_k$ , to estimate the dominant part of the forecast error covariance matrix  $\mathbf{P}_{k+1}^f$
  - (b) After using, the evolved perturbation  $\delta\mathbf{x}(t_{k-1})$  is discarded
  - (c) At the same time  $t_{k+1}$ , through the same assimilation scheme we calculate both the analysis state of the system  $\mathbf{x}_{k+1}^a$  and assimilate the perturbation  $\delta\mathbf{x}(t_k)$ , introduced the previous assimilation time  $t_k$  and evolved for all  $t \in ]t_k, t_{k+1}]$

- (d) Furthermore, also at time  $t_{k+1}$ , we randomly perturb the analysis state  $\mathbf{x}_{k+1}^a$  with a new, small vector  $\delta\mathbf{x}(t_{k+1})$ ; that's the *refresh*
2. For all  $t \in ]t_{k+1}, t_{k+2}]$ :
- (a) We evolve this new perturbation  $\delta\mathbf{x}(t_{k+1})$  with the same dynamics of the system, that is  $\mathbf{M} = \mathbf{M}(\mathbf{x}_{k+1}^a)$ , with the TLM operator  $\mathbf{M}$  evaluated along the forecast trajectory  $\mathbf{x}(t)$
  - (b) We evolve also  $\delta\mathbf{x}(t_k)$ , that was introduced at time  $t_k$  and that underwent to evolution for  $t \in ]t_k, t_{k+1}]$  as well as an assimilation procedure at the assimilation step  $t_{k+1}$  (see item 1.c)
  - (c) We let the new perturbation  $\delta\mathbf{x}(t_{k+1})$  grow for a suitable time before use, in this example for a breeding time  $\Delta t = 2\tau$ , so it will be used at assimilation time  $t_{k+3}$ . Since the time interval  $\Delta t$  is greater than the assimilation window  $\tau$ , the evolved perturbation  $\delta\mathbf{x}(t_{k+1})$  will undergo to the assimilation process at time  $t_{k+2}$ . The assimilation will be done by using the perturbation  $\delta\mathbf{x}(t_k)$ , evolved and yet assimilated at previous assimilation step  $t_{k+1}$
3. At assimilation step  $t_{k+2}$  the cycle is repeated

Throughout this work we will use a single bred vector  $\delta\mathbf{x}(t_k)$ . When it is ready, after a breeding time  $\Delta t = 2\tau$ , it is assumed to capture the most unstable structure of the forced system. It may be normalized or not, depending on the particular scheme adopted: if not normalized, an opportune initial length has to be chosen.



## Chapter 3

# Assimilation in the Lorenz 63 model: comparison among different methods

In the context of Lorenz convective 3-dimensional model (1963), we will run observation system simulation experiments in a perfect model setting and, in section 3.6, with some kind of model error as well.

In section 3.2, we will describe the algorithms used for different flavors of Extended Kalman Filters, while in section 3.3 we will do the same for different AUS assimilation schemes with increasing capabilities. We will then illustrate the results in section 3.4. In section 3.5 we will show some examples about the different behavior of the EKF and AUS assimilation schemes in some specific circumstances. Finally, in section 3.6, we will test our methods in the presence of model error.

### 3.1 Experimental setups

Since we are mainly interested in comparing different data assimilation schemes in critical circumstances, we choose hard setups for our experiments. Basically we will perform three kind of experiments, concerning:

- Synchronization to the “truth”
- Noisy observations
- Model error

In synchronization experiments we will qualitatively show for a few typical cases the capabilities to converge to the truth for the EKF, the Evensen's flavor of EKF (see section 3.2), and the best-performance AUS scheme (see section 3.3). In particular we will use a mix of perfect or quasi-perfect observations ( $\sigma^2 = 0$ ,  $\sigma^2 = 0.01$  or  $\sigma^2 = 0.1$ ) with long or very long assimilation windows ( $\tau = 0.25$  or  $\tau = 0.6$ ). In these case studies we will observe only the  $y$  variable, which is the most valuable one.

In noisy observation experiments the goal is to show the average RMS analysis and forecast error for all the different DA schemes with a long assimilation window ( $\tau = 0.25$ ) and noisy observations ( $\sigma^2 = 2$ ). The results shown are the mean values of 100,000 assimilations, with 3 or 2 variables observed. When 2 observations are used, the most valuable two ( $y$  and  $z$ ) are used for the EKF schemes, while for the AUS schemes we will use two adaptive observations (see subsection 2.3.5). Similar trials with different noisy observations ( $\sigma^2 = 1$  or  $\sigma^2 = 0.1$ ) will be performed, with 3 observed variables.

In model error experiments, first we will perform trials by adding a random error to the model equations at each integration step, then we will test the effects of systematic error by varying one model parameter. For these experiments the results shown are the mean values of 20,000 assimilations.

## 3.2 Extended Kalman Filters

When the EKF is applied to a strongly nonlinear system, such as Lorenz's three variables model, filter divergence can occur.

Different empirical techniques have been devised to overcome this difficulty: Evensen in 1997 added a term  $\mathbf{Q}$  akin to the model error covariance term in EKF [11]; Yang *et al.* in 2006 perturbed the analysis error covariance matrix and inflated the background error covariance matrix [36].

### 3.2.1 EKF

The assimilation window is set to  $\tau = 0.25$ , which is a quite large value for this dynamical system: due to the limits of the linear approximation, in these conditions the EKF tends to have poor performances with respect to other assimilation schemes, and often filter divergence occurs.

These experiments are based on standard EKF equations 2.76-2.80, with the covariance

matrix  $\mathbf{Q}$  set to zero, so:

$$\mathbf{x}_{k+1}^f = M\mathbf{x}_k^a \quad (3.1)$$

$$\mathbf{P}_{k+1}^f = M\mathbf{P}_k^a M^T \quad (3.2)$$

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1}^f \mathbf{H}^T (\mathbf{H} \mathbf{P}_{k+1}^f \mathbf{H}^T + \mathbf{R})^{-1} \quad (3.3)$$

$$\mathbf{x}_{k+1}^a = (\mathbf{I} - \mathbf{K}_{k+1} \mathbf{H}) \mathbf{x}_{k+1}^f + \mathbf{K}_{k+1} \mathbf{y}_{k+1}^o \quad (3.4)$$

$$\mathbf{P}_{k+1}^a = (\mathbf{I} - \mathbf{K}_{k+1} \mathbf{H}) \mathbf{P}_{k+1}^f \quad (3.5)$$

Here the observation operator is the same as its Jacobian  $H = \mathbf{H}$ . If we observe 3 variables at each assimilation step,  $\mathbf{H}$  will reduce to the identity  $3 \times 3$  matrix:

$$\mathbf{H}_{xyz} = \mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

If we observe 2 variables, they will be  $y$  and  $z$ , the most useful. The operator  $\mathbf{H}$  will be

$$\mathbf{H}_{yz} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.7)$$

### 3.2.2 Evensen's version of EKF

In Evensen [11], where the model is considered perfect, standard EKF equations 2.76-2.80 are still used, but the covariance matrix  $\mathbf{Q}$ , an additive term akin to the model error covariance, is used as a correction to avoid filter divergence. The matrix  $\mathbf{Q}$  has been estimated after optimization:

$$\mathbf{Q} = \begin{bmatrix} 0.1491 & 0.1505 & 0.0007 \\ 0.1505 & 0.9048 & 0.0014 \\ 0.0007 & 0.0014 & 0.9180 \end{bmatrix} \quad (3.8)$$

### 3.2.3 Yang's version of EKF

In Yang *et al.* (2006) the analysis error covariance has been perturbed and the background error covariance inflated [36]:

- *Random noise:* small random perturbations uniformly distributed between 0 and 1, and multiplied by  $\mu = 0.1$  (when 3 variables are observed) or  $\mu = 0.2$  (when 2 variables are

observed) are added to the diagonal of the analysis error covariance matrix obtained with the EKF at every assimilation step

- *Inflation*: the background error covariance is inflated by a factor of  $1 + \delta = 1.1$ , prior to the analysis step

Once the empirical parameters have been optimized for a specific assimilation interval and observation noise variance, these techniques provide satisfactory results, particularly for sufficiently short assimilation intervals.

### 3.3 Assimilation in the Unstable Subspace: further developments

In this section we will apply, to the L63 dynamical system, different formulations of the AUS scheme, with increasing capabilities. We use a single unstable vector of the analysis-forecast system, estimated by BDAS (see section 2.4). So the recurrence equations are [3]:

$$\mathbf{x}_{k+1}^f = M\mathbf{x}_k^a \quad (3.9)$$

$$\mathbf{P}_{k+1}^f = \gamma^2 \mathbf{e}_{k+1} \mathbf{e}_{k+1}^T \quad (3.10)$$

$$\mathbf{K}_{k+1} = \gamma_{k+1}^2 \mathbf{e}_{k+1} (\mathbf{H}\mathbf{e}_{k+1})^T [\gamma_{k+1}^2 (\mathbf{H}\mathbf{e}_{k+1})(\mathbf{H}\mathbf{e}_{k+1})^T + \mathbf{R}]^{-1} \quad (3.11)$$

$$\mathbf{x}_{k+1}^a = (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H})\mathbf{x}_{k+1}^f + \mathbf{K}_{k+1}\mathbf{y}_{k+1}^o \quad (3.12)$$

$$\mathbf{P}_{k+1}^a = (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H})\mathbf{P}_{k+1}^f \quad (3.13)$$

where the symbols have the usual meaning as in EKF equations, except for:

- the normalized column vector  $\mathbf{e}_{k+1}$ , that is the single unstable unit vector, at assimilation step  $t_{k+1}$ , estimated by BDAS. The bulk of the forecast error covariance matrix  $\mathbf{P}_{k+1}^f$  is computed in the unstable 1-dimensional subspace defined by the unit vector  $\mathbf{e}_{k+1}$ , and assumed to be small in the complementary subspace
- the amplitude of the forecast error  $\gamma$

Note also that here the observation operator  $H$  is actually linear (we simply observe a variable or not) and it has then been replaced by its linearized version:  $H = \mathbf{H}$ . The analysis covariance matrix  $\mathbf{P}_{k+1}^a$  (eq. 3.13) has been shown here for completeness, but it's not actually used in the estimate of the successive  $\mathbf{P}_{k+2}^f$ .



A key problem in AUS is the estimate of the amplitude of the forecast error  $\gamma$ . We begin in subsection 3.3.1 with a very basic AUS- $\gamma_0$  approach, where no use of observations is made in the estimate of  $\gamma$ , then in subsection 3.3.2 we introduce a new estimate of  $\gamma$  from observations, that will be used in subsection 3.3.3. Then a new formulation of AUS is introduced in subsections 3.3.4 and 3.3.5.

### 3.3.1 AUS- $\gamma_0$ : no use of observations in the estimate of the forecast error amplitude

This DA scheme is a simplified formulation of the AUS approach. The main reason why we implement this assimilation scheme is to show that we actually need a way to estimate the amplitude  $\gamma$  of the forecast error in the unstable subspace from observations. In this assimilation scheme, instead, we simply optimize the initial amplitude  $\alpha$  used in the re-normalization of the random perturbation. By breeding on the data-assimilation system (BDAS), we obtain the perturbation in the unstable 1-dimensional subspace, whose amplitude is  $\gamma$ , at the time it is used in the assimilation. During the forecast steps this perturbation will amplify and its amplitude will be reduced at assimilation time.

Let's suppose that the breeding time has been chosen equal to 2 assimilation windows: the unstable vector may need more time than a single assimilation window to grow and to capture the instability of the flow. So, at each assimilation step  $t_{k+1}$  (see Table 2.1):

- we use in the assimilation process the vector  $\gamma \mathbf{e}_{k+1}$ , i.e. the evolution, through the tangent linear model propagator, of the random perturbation  $\alpha \delta \mathbf{x}_{k-1}$  introduced two assimilation steps earlier, that underwent also the assimilation process in the previous step  $t_k$  through the  $(\mathbf{I} - \mathbf{K}_k \mathbf{H})$  operator
- in the same way, we assimilate the observations in the trajectory obtained by adding  $\alpha \delta \mathbf{x}_k$  to the control at the previous assimilation step. The evolved perturbation will be used at the next assimilation step, time  $t_{k+2}$ , to estimate  $\mathbf{P}_{k+2}^f$
- we introduce a new perturbation  $\alpha \delta \mathbf{x}_{k+1}$ , where  $\delta \mathbf{x}_{k+1}$  is a random vector whose components are Gaussian, with zero mean and standard deviation 1; its evolution will be used at assimilation step  $t_{k+3}$ , after having undergone the assimilation process at time  $t_{k+2}$

The parameter  $\alpha$  needs to be tuned. In the present application the optimized value is  $\alpha = 1$ . The estimate of the forecast error covariance matrix is

$$\mathbf{P}_{k+1}^f = \gamma^2 \mathbf{e}_{k+1} \mathbf{e}_{k+1}^T \quad (3.14)$$

The Kalman gain, in the same spirit as in the EKF, is

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1}^f \mathbf{H}^T (\mathbf{H} \mathbf{P}_{k+1}^f \mathbf{H}^T + \mathbf{R})^{-1} \quad (3.15)$$

It will be used to estimate the analysis state, by correcting the forecast state, via the

$$\mathbf{x}_{k+1}^a = (\mathbf{I} - \mathbf{K}_{k+1} \mathbf{H}) \mathbf{x}_{k+1}^f + \mathbf{K}_{k+1} \mathbf{y}_{k+1}^o \quad (3.16)$$

and the analysis covariance matrix

$$\mathbf{P}_{k+1}^a = (\mathbf{I} - \mathbf{K}_{k+1} \mathbf{H}) \mathbf{P}_{k+1}^f \quad (3.17)$$

The unstable vector  $\delta \mathbf{x}_k$ , introduced at the previous assimilation step  $t_k$ , will be corrected as well:

$$\delta \mathbf{x}_k^a = (\mathbf{I} - \mathbf{K}_{k+1} \mathbf{H}) \mathbf{M} \delta \mathbf{x}_{k-1} \quad (3.18)$$

### 3.3.2 Estimate of the amplitude $\gamma$ of the forecast error from observations

An estimate of the amplitude  $\gamma$  of the forecast error in a single unstable direction  $\mathbf{e}$  can be obtained in the following way, by rearranging the definition of the forecast error and by assuming that the bulk of it is along the vector  $\mathbf{e}$ :

$$\mathbf{H} \boldsymbol{\eta}^f = \mathbf{H}(\mathbf{x}^f - \mathbf{x}^t) = -\mathbf{d} + \boldsymbol{\eta}^o \quad (3.19)$$

$$\mathbf{x}^f - \mathbf{x}^t = \gamma \mathbf{e} + \delta \mathbf{f} \quad (3.20)$$

$$\mathbf{H}(\mathbf{x}^f - \mathbf{x}^t) = \gamma \mathbf{H} \mathbf{e} + \delta \mathbf{H} \mathbf{f} \quad (3.21)$$

where we dropped the subscripts  $k+1$  referring to the assimilation time and where

$$\begin{aligned} \mathbf{d} &= \mathbf{y}^o - \mathbf{H} \mathbf{x}^f && \text{innovation} \\ \boldsymbol{\eta}^o &= \mathbf{y}^o - \mathbf{H} \mathbf{x}^t && \text{observational error} \\ \boldsymbol{\eta}^f &= \mathbf{x}^f - \mathbf{x}^t && \text{forecast error} \end{aligned}$$

The vector  $\mathbf{f}$  spans the subspace complementary of the unstable subspace. So:

$$\gamma \mathbf{H} \mathbf{e} + \delta \mathbf{H} \mathbf{f} = -\mathbf{d} + \boldsymbol{\eta}^o \quad (3.22)$$

Now, if we left-multiply by  $\mathbf{e}^T \mathbf{H}^T$ :

$$\gamma \mathbf{e}^T \mathbf{H}^T \mathbf{H} \mathbf{e} + \delta \mathbf{e}^T \mathbf{H}^T \mathbf{H} \mathbf{f} = -\mathbf{e}^T \mathbf{H}^T \mathbf{d} + \mathbf{e}^T \mathbf{H}^T \boldsymbol{\eta}^o \quad (3.23)$$

Neglecting the terms  $\delta \mathbf{e}^T \mathbf{H}^T \mathbf{H} \mathbf{f}$  and  $\mathbf{e}^T \mathbf{H}^T \boldsymbol{\eta}^o$  (which is zero, on average), we obtain the following estimate of  $\gamma$ :

$$\gamma \simeq -(\mathbf{e}^T \mathbf{H}^T \mathbf{H} \mathbf{e})^{-1} \mathbf{e}^T \mathbf{H}^T \mathbf{d} \quad (3.24)$$

Our estimate of  $\mathbf{P}^f$  is the same as in eq. 2.106, but with this new estimate of  $\gamma$ :

$$\mathbf{P}^f \simeq \gamma^2 \mathbf{e} \cdot \mathbf{e}^T \quad (3.25)$$

This new estimate of  $\gamma$  also affects the corresponding gain matrix (see eq. 2.107):

$$\mathbf{K} \simeq \gamma^2 \mathbf{e} \mathbf{e}^T \mathbf{H}^T (\gamma^2 \mathbf{H} \mathbf{e} \mathbf{e}^T \mathbf{H}^T + \mathbf{R})^{-1} \quad (3.26)$$

Accordingly, the analysis increment of eq. 2.95 becomes:

$$\delta \mathbf{x}^a = \mathbf{x}^a - \mathbf{x}^f = \mathbf{K} (\mathbf{y}^o - \mathbf{H} \mathbf{x}^f) \quad (3.27)$$

$$= \gamma^2 \mathbf{e} \mathbf{e}^T \mathbf{H}^T (\gamma^2 \mathbf{H} \mathbf{e} \mathbf{e}^T \mathbf{H}^T + \mathbf{R})^{-1} (\mathbf{y}^o - \mathbf{H} \mathbf{x}^f) \quad (3.28)$$

$$= \mathbf{e} c \quad (3.29)$$

In equation 3.29,  $c$  is a scalar coefficient.

When the assimilation interval is sufficiently long with respect to the typical doubling time and the forecast error becomes large with respect to the observation error, the estimate of  $\gamma$  from observations leads to significant improvement of assimilation performance.

### 3.3.3 AUS- $\gamma$ : using the estimate of $\gamma$ from observations

In this scheme we use the same approach as in AUS- $\gamma_0$ , but with three important improvements:

- The unstable direction  $\mathbf{e}$  (normalized to 1), which is again estimated via BDAS, is estimated by evolving an infinitesimal perturbation  $\delta \mathbf{x}_{k-1}$ , so that we can expect the perturbed trajectory to still be on the attractor (or at least not far from it)
- An estimate of the amplitude of the forecast error  $\gamma$  from observations, as described in

subsection 3.3.2; the amplitude of the forecast error is intended to be in the unstable direction  $\mathbf{e}$

- A new estimate of  $\mathbf{P}_{k+1}^f$

The second and third improvements have been discussed in subsection 3.3.2: after estimating the amplitude of the forecast error  $\gamma$  from observations, the forecast error covariance matrix will be

$$\mathbf{P}_{k+1}^f = \gamma^2 \mathbf{e}_{k+1} \mathbf{e}_{k+1}^T \quad (3.30)$$

The gain matrix will be computed with

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1}^f \mathbf{H}^T (\mathbf{H} \mathbf{P}_{k+1}^f \mathbf{H}^T + \mathbf{R})^{-1} \quad (3.31)$$

which is formally the same as eq. 3.15, but the better estimate of  $\mathbf{P}_{k+1}^f$  will in turn provide the following better approximated expression, already mentioned in eq. 3.26:

$$\mathbf{K}_{k+1} = \gamma^2 \mathbf{e}_{k+1} \mathbf{e}_{k+1}^T \mathbf{H}^T (\gamma^2 \mathbf{H} \mathbf{e}_{k+1} \mathbf{e}_{k+1}^T \mathbf{H}^T + \mathbf{R})^{-1}$$

We are now able to compute the analysis state  $\mathbf{x}_{k+1}^a$  and the analysis covariance matrix  $\mathbf{P}_{k+1}^a$  through equations 3.16 and 3.17, as usual.

It will be shown in section 3.4 that this algorithm will greatly enhance the performance of AUS scheme.

### 3.3.4 Iterating

This assimilation scheme is a further improvement to AUS- $\gamma$ . After calculating the coefficient  $c$  at time  $t_{k+1}$  (eq. 3.29), we apply at the previous analysis state  $\mathbf{x}_k^a$  the perturbation  $\Delta \mathbf{x}_k^a = c \exp(-\int_{t_k}^{t_{k+1}} \lambda(t) dt) \mathbf{e}_k$  that, if the error behaved linearly, would lead exactly to the analysis  $\mathbf{x}_{k+1}^a$  obtained from the analysis increment expression (3.27-3.29). In fact, from time  $t_k$  to  $t_{k+1}$ , we have:

$$\mathbf{M} \mathbf{e}_k = \exp \left( \int_{t_k}^{t_{k+1}} \lambda(t) dt \right) \mathbf{e}_{k+1} \quad (3.32)$$

where  $\lambda(t)$  is the leading local Lyapunov exponent. Therefore:

$$\mathbf{M} \Delta \mathbf{x}_k^a = c \exp \left( - \int_{t_k}^{t_{k+1}} \lambda(t) dt \right) \mathbf{M} \mathbf{e}_k \quad (3.33)$$

$$= c \exp \left( - \int_{t_k}^{t_{k+1}} \lambda(t) dt \right) \exp \left( \int_{t_k}^{t_{k+1}} \lambda(t) dt \right) \mathbf{e}_{k+1} \quad (3.34)$$

$$= c \mathbf{e}_{k+1} \quad (3.35)$$

We now integrate the system nonlinearly from this new estimate of the previous analysis state  $\mathbf{x}_k^a$ , evolve the perturbation  $\mathbf{e}_k$  following the updated nonlinear trajectory from  $t_k$  to  $t_{k+1}$  and perform the final analysis at time  $t_{k+1}$ . This procedure is hereafter referred to as **AUS-iterating**: it is a major improvement with respect to the AUS- $\gamma$  scheme.

### 3.3.5 Iterating and using a quasi-static $\mathbf{P}^f$ in stable zones of the attractor

This technique is a refinement of AUS-iterating. If the dynamical system is passing through a zone of the attractor where there are stable trajectories, i.e. where they do not diverge, we can assume that errors do not actually grow.

So, in these stable situations (where the amplification factor during forecast step is  $\leq 1$ ) we use a quasi-static, diagonal  $\mathbf{P}^f$ . If:

$$\|\mathbf{M} \delta \mathbf{x}_k\| < \|\delta \mathbf{x}_k\| \quad (3.36)$$

then we set:

$$\mathbf{P}_{k+1}^f = a^2 \left\{ \frac{\|\mathbf{M} \delta \mathbf{x}_k\|}{\|\delta \mathbf{x}_k\|} \right\}^2 \mathbf{I} \quad (3.37)$$

where  $\mathbf{M} \delta \mathbf{x}_k$  is the evolution from time  $t_k$  to  $t_{k+1}$  of the perturbation  $\delta \mathbf{x}_k$ , which in turn is the evolution of a random perturbation, introduced at time  $t_{k-1}$ , which has grown from time  $t_{k-1}$  to  $t_k$  and undergone an assimilation process at time  $t_k$ . The coefficient  $a^2$  is proportional to the square of the average analysis error. In particular, it has been chosen

$$a^2 = \frac{1}{2} < \eta^a >^2 \quad (3.38)$$

where  $\eta^a$  is the analysis error. This procedure is hereafter referred to as **AUS-iterating+**: it provides the best performances.

## 3.4 Comparing results

In this section we will compare the performances of the different assimilation schemes under investigation. In the following remarks we will use the acronyms of Table 3.1.

We will begin with some case studies of synchronization to the truth with a perfect or quasi-perfect observation (the variable  $y$ ) and a long assimilation window; then we will deal with

Table 3.1: The different data assimilation schemes under investigation with their main features.

|                 |  |
|-----------------|--|
| EKF             | pure EKF   |
| EKF-Evensen     | Evensen's EKF, with an optimized $\mathbf{Q}$ [11]   |
| EKF-Yang        | Yang's EKF, with random noise in $\mathbf{P}^a$ and inflation [36]   |
| AUS- $\gamma_0$ | AUS with no use of observations for the estimate of the amplitude of the forecast error $\gamma$ ; $\mathbf{P}^f = \gamma^2 \mathbf{e} \mathbf{e}^T$                 |
| AUS- $\gamma$   | AUS with $\gamma = -(\mathbf{e}^T \mathbf{H}^T \mathbf{H} \mathbf{e})^{-1} \mathbf{e}^T \mathbf{H}^T \mathbf{d}$ , $\mathbf{P}^f = \gamma^2 \mathbf{e} \mathbf{e}^T$ |
| AUS-iterating   | AUS- $\gamma$ with iterations  |
| AUS-iterating+  | AUS-iterating with static $\mathbf{P}_{k+1}^f$ if amplification factor $\leq 1$ during forecast step   |

shorter assimilation windows and noisy observations in different configurations.

### 3.4.1 Synchronization: one perfect or quasi-perfect observation, case studies

In this subsection we present a few case studies about synchronization with truth for perfect or quasi-perfect observations, for different assimilation windows. The case studies shown have been chosen for the clarity of the resulting plots and because they are neither particularly advantageous nor pathological for any assimilation scheme. We're going to compare only EKF, EKF-Evensen and AUS-iterating assimilation schemes, which do not need any tuning of parameters. We will notice that all of them eventually converge to the truth, but AUS-iterating converges faster than EKF or EKF-Evensen. Furthermore, for critical situations, in terms of assimilation window length or observational error amplitude, AUS-iterating still converges to the truth in cases when the EKF or EKF-Evensen do not.

It's known that, even with a single perfect observation only, the EKF may synchronize with the truth, if the assimilation window is not too long. This is shown in Fig. 3.1, where we observe  $y$  with no observational error ( $\sigma^2 = 0$ ) with assimilation window  $\tau = 0.25$ . You can notice how AUS-iterating DA scheme converges much faster.

The pure EKF and the EKF-Evensen version give the same results for  $\sigma^2 = 0$ , since the gain matrix and the analysis vector will reduce in both cases to the same simpler expression.

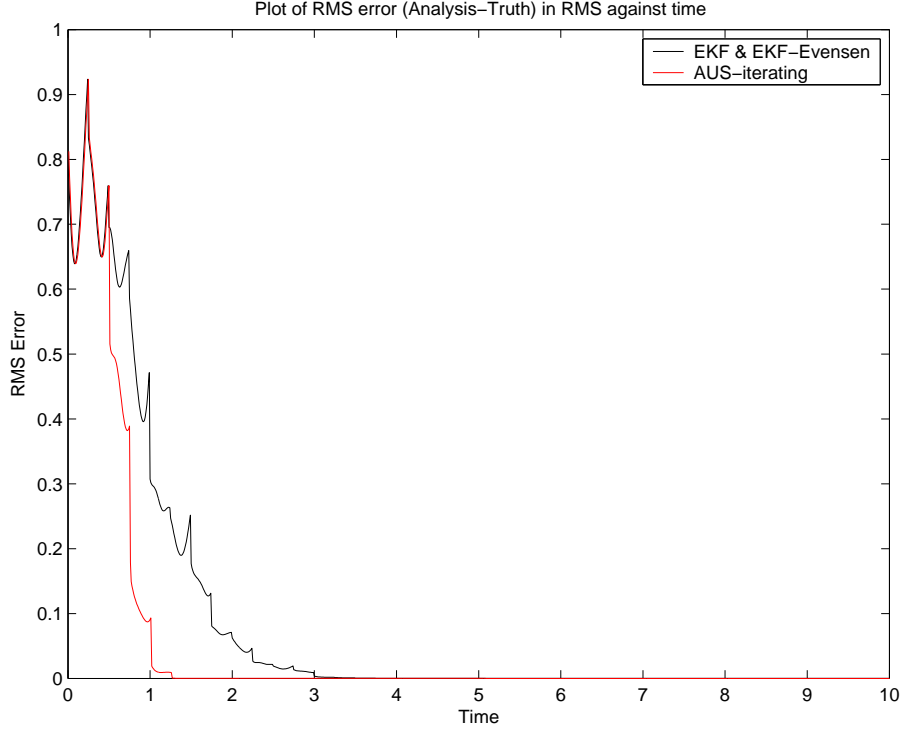


Figure 3.1: Synchronization with truth for both EKF/EKF-Evensen and AUS-iterating assimilation schemes. The only observed variable is  $y$  with variance  $\sigma^2 = 0$  and assimilation window  $\tau = 0.25$

From the Kalman gain expression (eq. 2.78):

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1}^f \mathbf{H}^T (\mathbf{H} \mathbf{P}_{k+1}^f \mathbf{H}^T + \mathbf{R})^{-1}$$

if the matrix  $\mathbf{R} = \mathbf{0}$ , in observation space:

$$\mathbf{H} \mathbf{K}_{k+1} = \mathbf{H} \mathbf{P}_{k+1}^f \mathbf{H}^T (\mathbf{H} \mathbf{P}_{k+1}^f \mathbf{H}^T)^{-1} \quad (3.39)$$

$$= \mathbf{I} \quad (3.40)$$

Thus, from the analysis state expression (eq. 2.79):

$$\mathbf{x}_{k+1}^a = (\mathbf{I} - \mathbf{K}_{k+1} \mathbf{H}) \mathbf{x}_{k+1}^f + \mathbf{K}_{k+1} \mathbf{y}_{k+1}^o$$

we get:

$$\mathbf{H} \mathbf{x}_{k+1}^a = \mathbf{y}_{k+1}^o \quad (3.41)$$

Just for example, in Figures 3.2-3.5 the EKF and EKF-Evensen solution are shown for  $x$  and the assimilation performance for  $\sigma^2 = 0$  and assimilation window  $\tau = 0.6$ . The vertical dotted

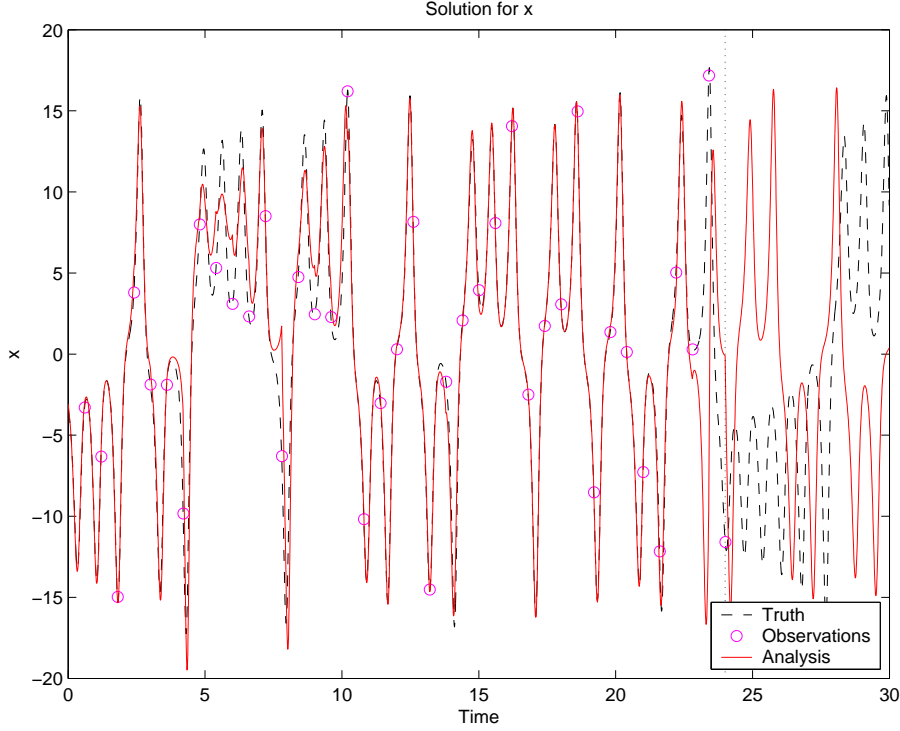


Figure 3.2: EKF and EKF-Evensen (same plot in these conditions, see text and eq. 3.40) find it hard to synchronize with truth. The only observed variable is  $y$  with variance  $\sigma^2 = 0$  and assimilation window  $\tau = 0.6$

line is the time of the last observation. In Figures 3.6-3.10 the same is shown for  $\sigma^2 = 0.01$  and assimilation window  $\tau = 0.6$  while Figures 3.11-3.15 show the case with  $\sigma^2 = 0.1$  and assimilation window  $\tau = 0.25$ .

We can see that, when synchronization occurs for all DA schemes, AUS-iterating has the most rapid convergence to the truth; when it does not occur, AUS-iterating has an overall better performance. Furthermore, there are circumstances in which AUS-iterating synchronizes to the truth while neither EKF nor EKF-Evensen do.

### 3.4.2 Noisy observations with variance $\sigma^2 = 2$

These experiments are performed with the following setup:

- A  $10^5$  assimilations statistics
- 3 or 2 noisy observations at each assimilation step, with variance  $\sigma^2 = 2 \Rightarrow \sigma = 1.414$
- an assimilation window  $\tau = 0.25$



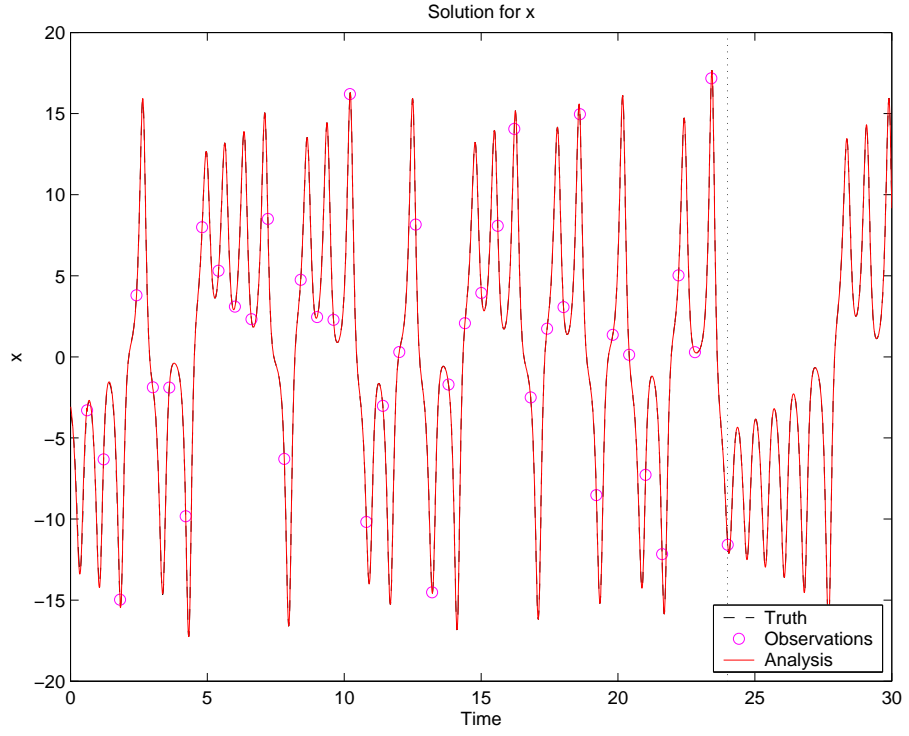


Figure 3.3: AUS-iterating: synchronization with truth. The only observed variable is  $y$  with variance  $\sigma^2 = 0$  and assimilation window  $\tau = 0.6$

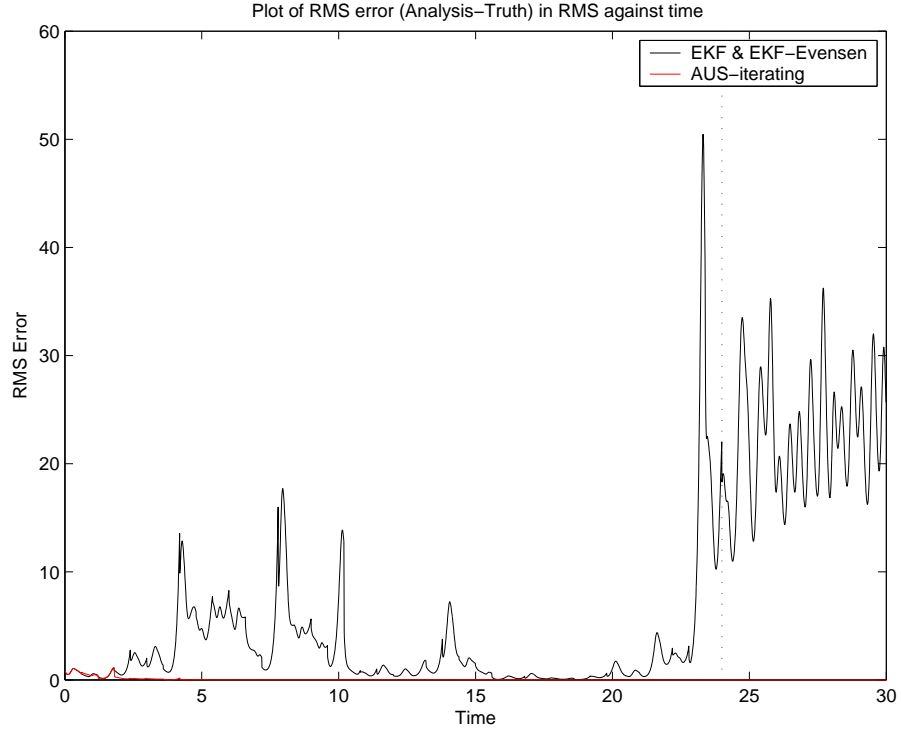


Figure 3.4: EKF/EKF-Evensen and AUS-iterating RMS errors. While the former fail to converge to the truth, the latter synchronizes very quickly. The only observed variable is  $y$  with variance  $\sigma^2 = 0$  and assimilation window  $\tau = 0.6$

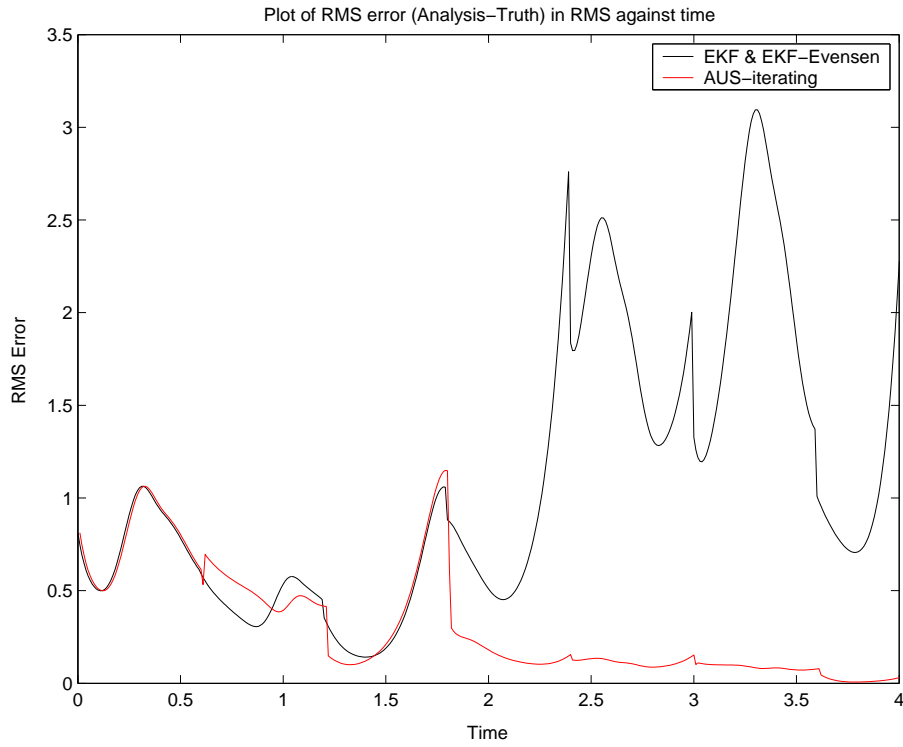


Figure 3.5: EKF/EKF-Evensen and AUS-iterating RMS errors: a zoom of the previous Fig. 3.4.

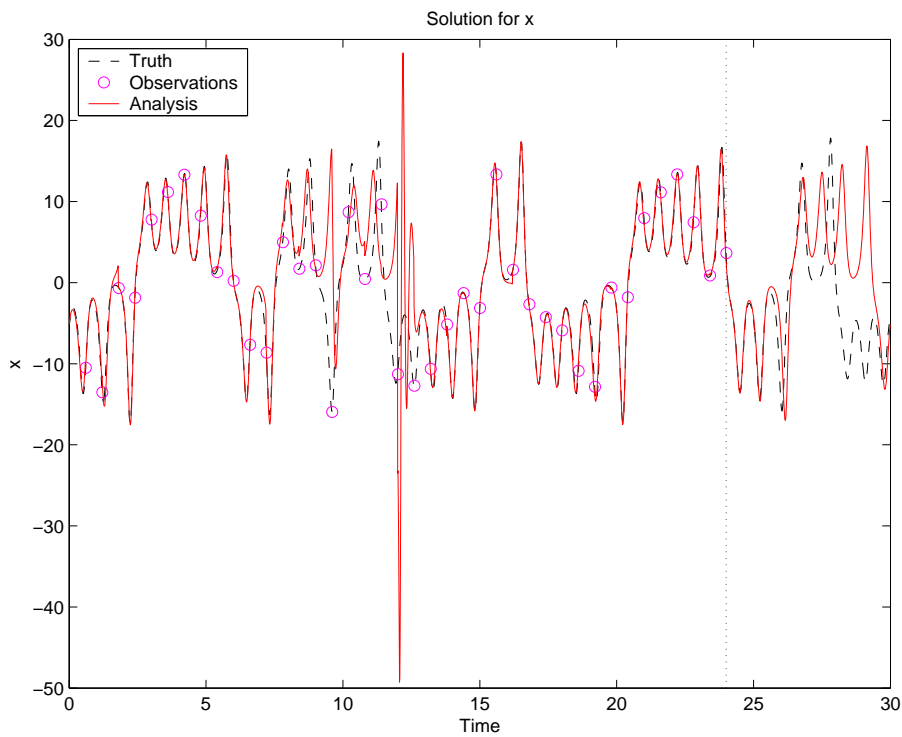


Figure 3.6: EKF: the only observed variable is  $y$  with variance  $\sigma^2 = 0.01$  and assimilation window  $\tau = 0.6$ ; in this case and in these conditions the EKF fails to synchronize to the truth. Note also the bad performance around time 14.

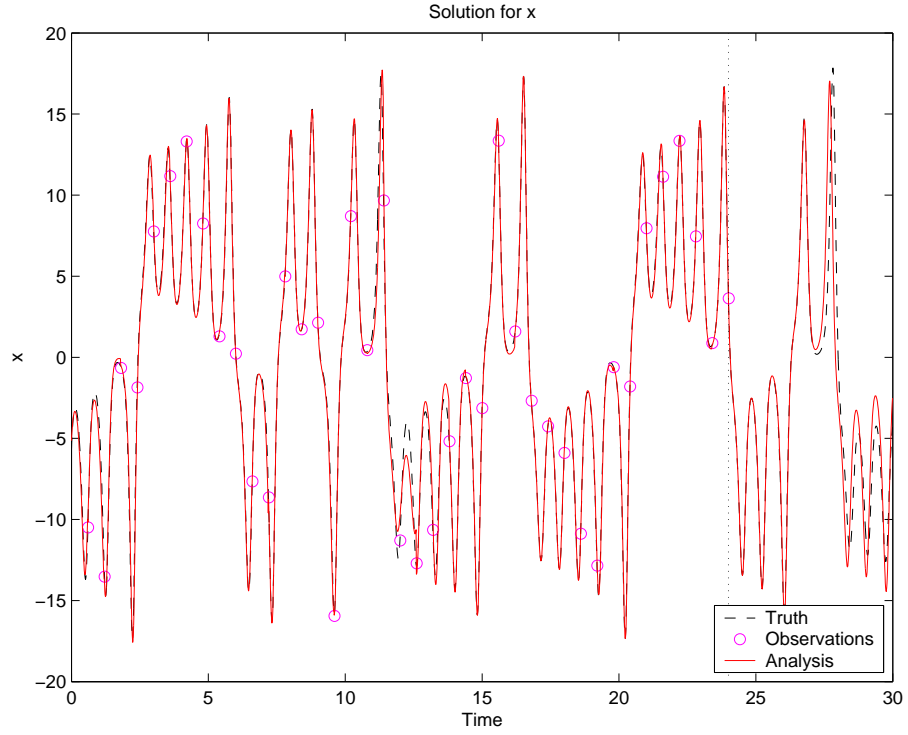


Figure 3.7: EKF-Evensen: the only observed variable is  $y$  with variance  $\sigma^2 = 0.01$  and assimilation window  $\tau = 0.6$ . A better performance than pure EKF.

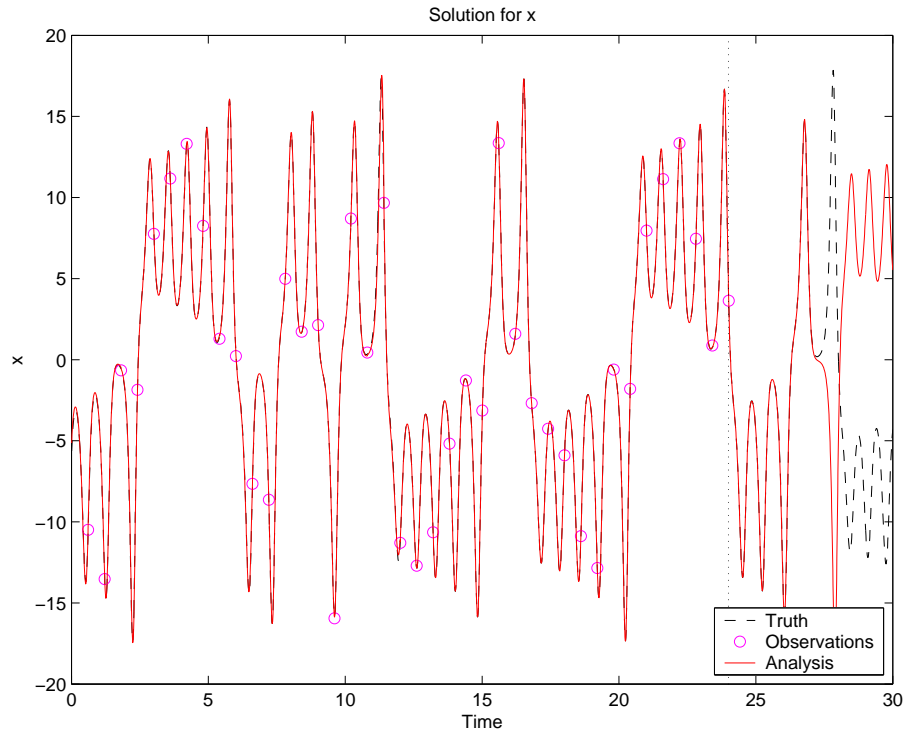


Figure 3.8: AUS-iterating: the only observed variable is  $y$  with variance  $\sigma^2 = 0.01$  and assimilation window  $\tau = 0.6$ . A far better performance.

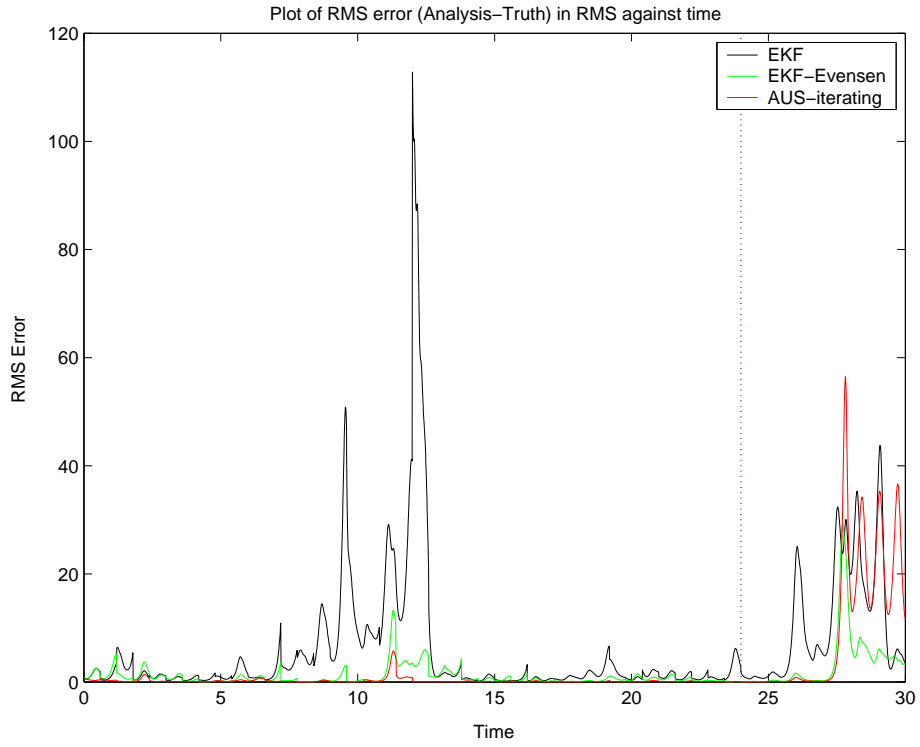


Figure 3.9: EKF, EKF-Evensen and AUS-iterating RMS errors: respective performances. The only observed variable is  $y$  with variance  $\sigma^2 = 0.01$  and assimilation window  $\tau = 0.6$

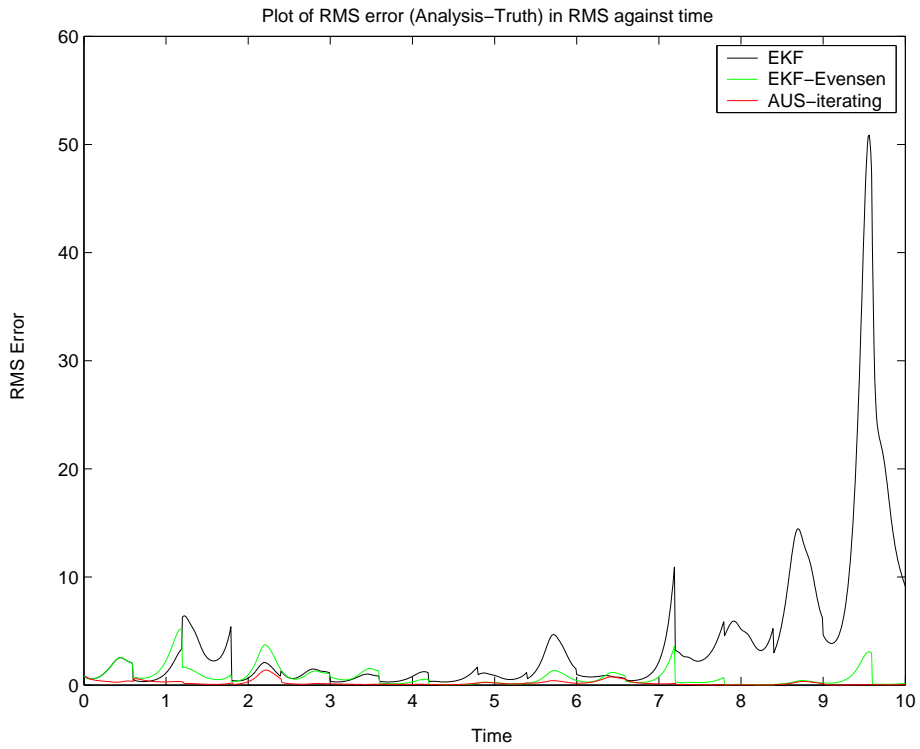


Figure 3.10: EKF, EKF-Evensen and AUS-iterating RMS errors: a zoom of the previous Fig. 3.9.

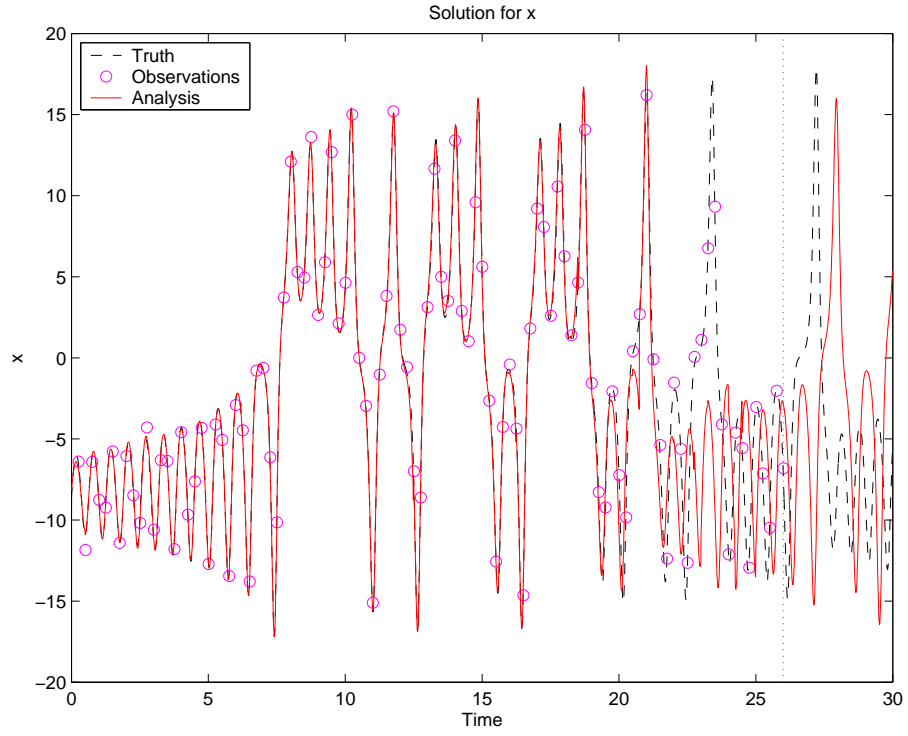


Figure 3.11: EKF does not synchronize with truth. The only observed variable is  $y$  with variance  $\sigma^2 = 0.1$  and assimilation window  $\tau = 0.25$

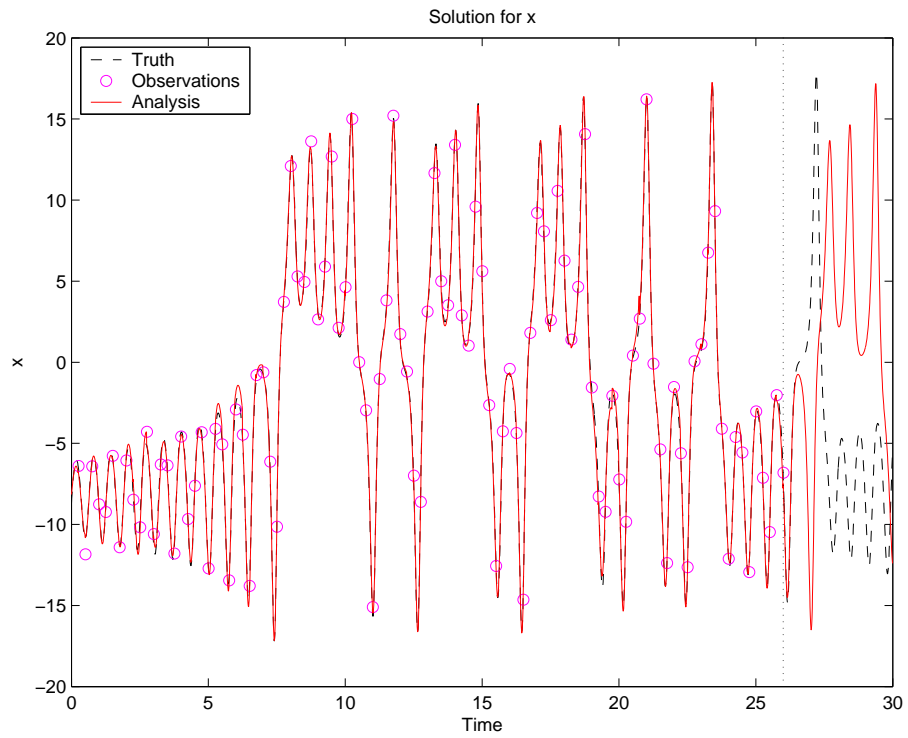


Figure 3.12: EKF-Evensen does not synchronize with truth. The only observed variable is  $y$  with variance  $\sigma^2 = 0.1$  and assimilation window  $\tau = 0.25$

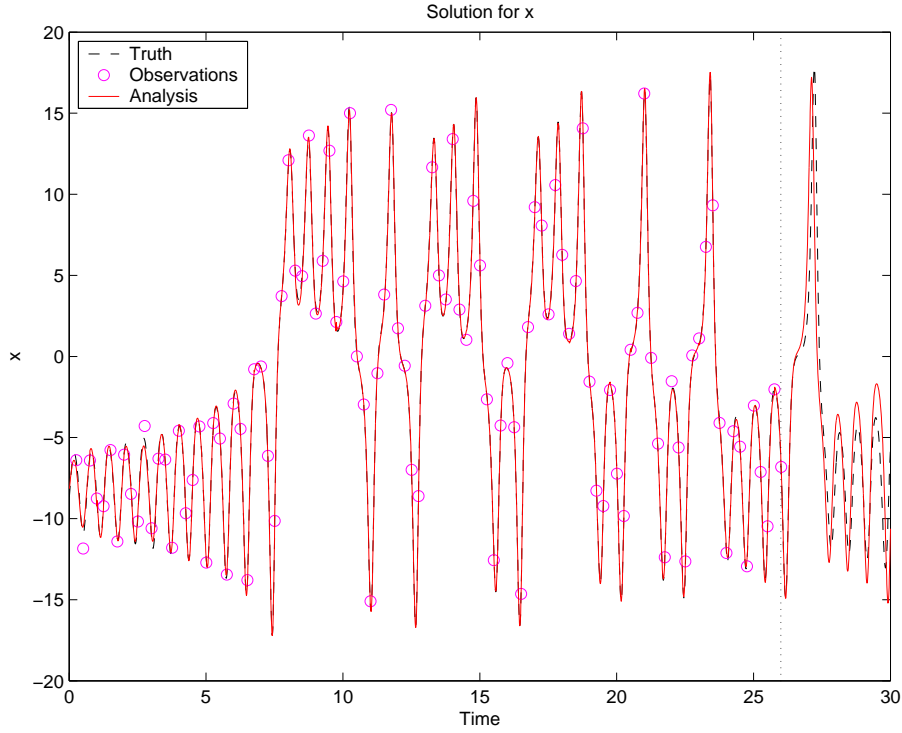


Figure 3.13: AUS-iterating does not synchronize with truth. The only observed variable is  $y$  with variance  $\sigma^2 = 0.1$  and assimilation window  $\tau = 0.25$

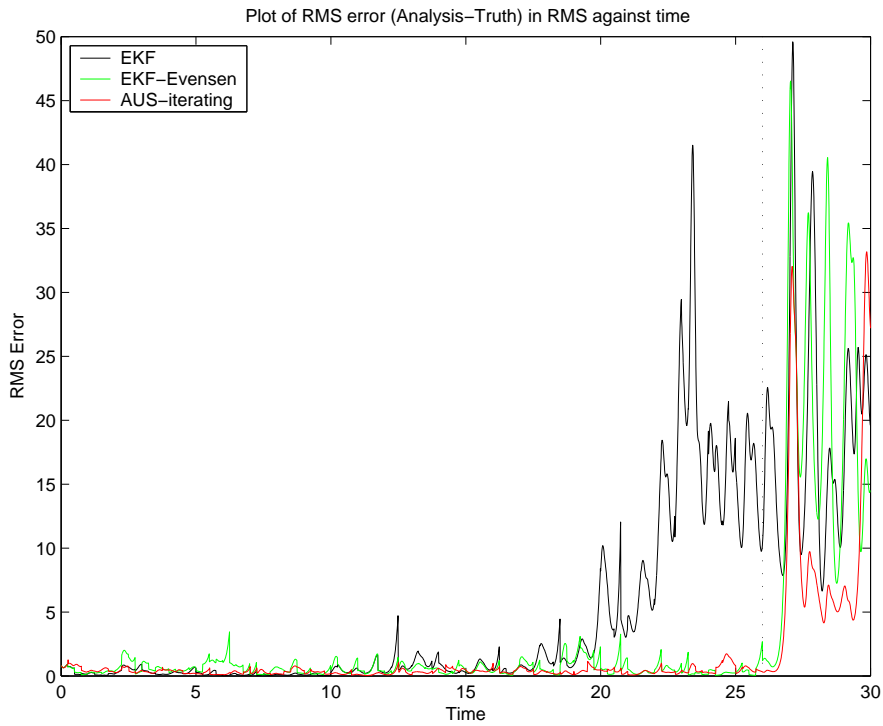


Figure 3.14: The only observed variable is  $y$  with variance  $\sigma^2 = 0.1$  and assimilation window  $\tau = 0.25$ : in these conditions, no assimilation scheme under investigation actually converge to the truth, but AUS-iterating has a better performance than EKF-Evensen, which in turn is far better than pure EKF.

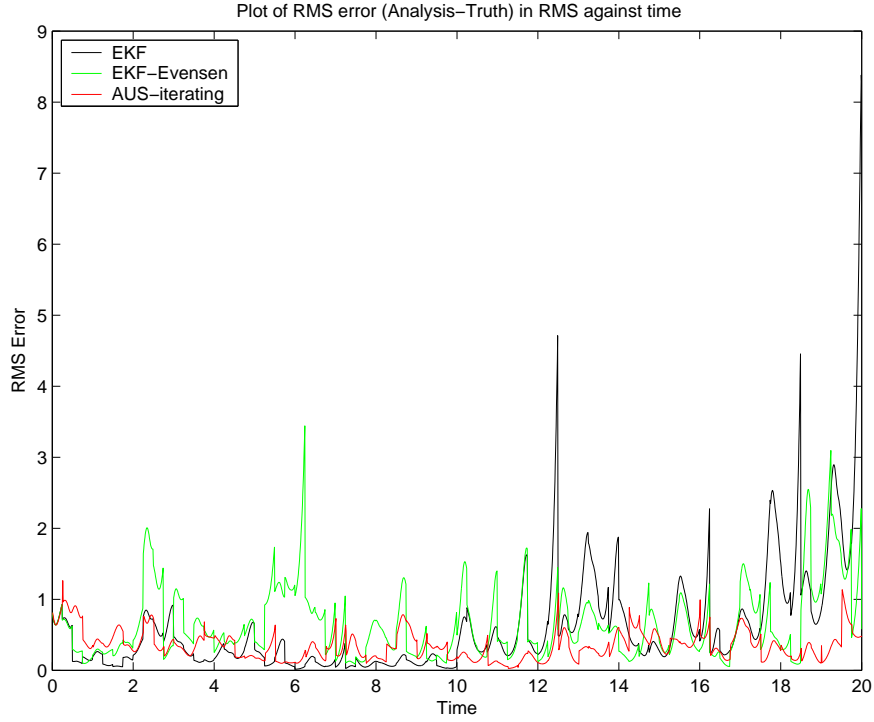


Figure 3.15: A zoom of the previous Fig. 3.14. Even if the global performance of EKF is poor because of filter divergence, this does *not* mean that EKF is *always* worse than others DA schemes for *all* times; while the global performance of AUS-iterating is globally better.

When 2 observations are used, the most valuable two ( $y$  and  $z$ ) are used for the EKF schemes; for the AUS schemes two adaptive observations are used (see subsection 2.3.5). In Figures 3.16-3.20 we show the average RMS analysis error distributions for 3 and 2 observed variables, and the forecast error distribution at time  $T+0.25$ ,  $T+0.5$  and  $T+0.75$ , where  $T$  is the assimilation time. All distributions are simply truncated at RMS error equal to 10.

These distributions show that not only AUS-iterating schemes are better — on average — than competitors, but their right tails are much less populated, too: the regime's changes tracking capability has been highly improved.

Numerical results are also shown in Tables 3.2 and 3.3: we can see that the AUS-iterating schemes outperform the other techniques, with even an average RMS analysis error well below the observations standard deviation. Similar relative performances can be seen for the average RMS forecast error. A further conclusion may be drawn: from the results of AUS- $\gamma_0$  and AUS- $\gamma$  schemes, we see that the proposed estimate of the forecast error amplitude from observations is really helpful, since it greatly boosts the assimilation performance.

Table 3.2: RMS analysis error, an average over 100,000 assimilations. 3 and 2 noisy observations with variance  $\sigma^2 = 2 \Rightarrow \sigma = 1.414$ . Assimilation window  $\tau = 0.25$ .

| <i>assimilation technique</i> | $\tau = 0.25, 3 \text{ obs}$ | $\tau = 0.25, 2 \text{ obs}$ |
|-------------------------------|------------------------------|------------------------------|
| EKF                           | 15.5                         | 15.6                         |
| EKF-Evensen                   | 1.72                         | 1.79                         |
| EKF-Yang                      | 3.77                         | 3.90                         |
| AUS- $\gamma_0$               | 7.02                         | 7.37                         |
| AUS- $\gamma$                 | 2.27                         | 2.52                         |
| AUS-iterating                 | 1.38                         | 1.58                         |
| AUS-iterating+                | 1.16                         | 1.33                         |

Table 3.3: RMS forecast error, an average over 100,000 assimilations. The mean RMS analysis error is the same as in Table 3.2 and is shown here again for comparison. 3 noisy observations with variance  $\sigma^2 = 2 \Rightarrow \sigma = 1.414$ . Assimilation window  $\tau = 0.25$ .

| <i>Assimilation technique</i> | <RMS analysis error> | <RMS forecast error> |          |          |
|-------------------------------|----------------------|----------------------|----------|----------|
|                               |                      | @ T+0.25             | @ T+0.50 | @ T+0.75 |
| EKF                           | 15.5                 | 16.6                 | 16.9     | 16.9     |
| EKF-Evensen                   | 1.72                 | 3.27                 | 6.26     | 7.24     |
| EKF-Yang                      | 3.77                 | 5.36                 | 6.97     | 7.64     |
| AUS- $\gamma_0$               | 7.02                 | 8.81                 | 10.4     | 11.1     |
| AUS- $\gamma$                 | 2.27                 | 3.79                 | 6.23     | 7.33     |
| AUS-iterating                 | 1.38                 | 2.75                 | 4.94     | 5.61     |
| AUS-iterating+                | 1.16                 | 2.33                 | 4.06     | 4.71     |



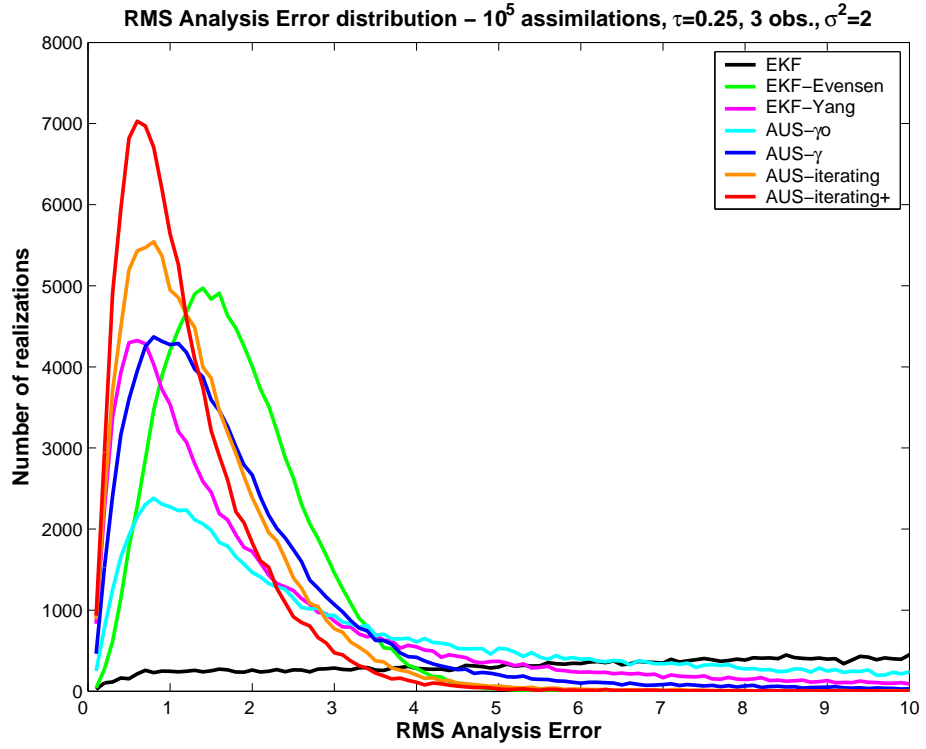


Figure 3.16: RMS Analysis Error distribution: an average on 100,000 assimilations with an assimilation window  $\tau = 0.25$ , 3 noisy observations with variance  $\sigma^2 = 2$

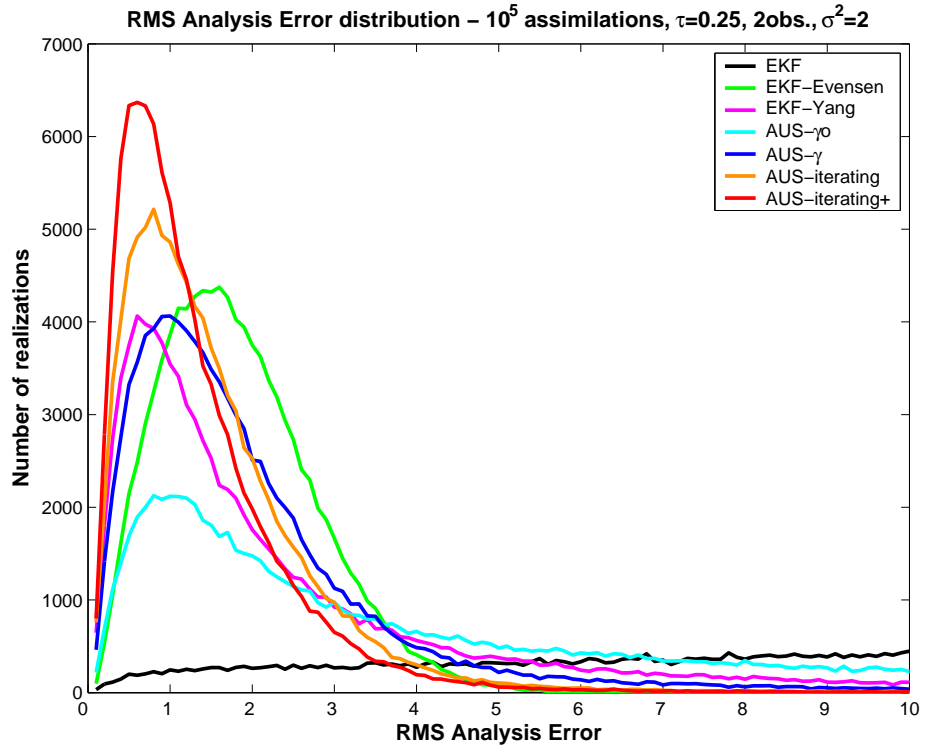


Figure 3.17: RMS Analysis Error distribution: an average on 100,000 assimilations with an assimilation window  $\tau = 0.25$ , 2 noisy observations with variance  $\sigma^2 = 2$

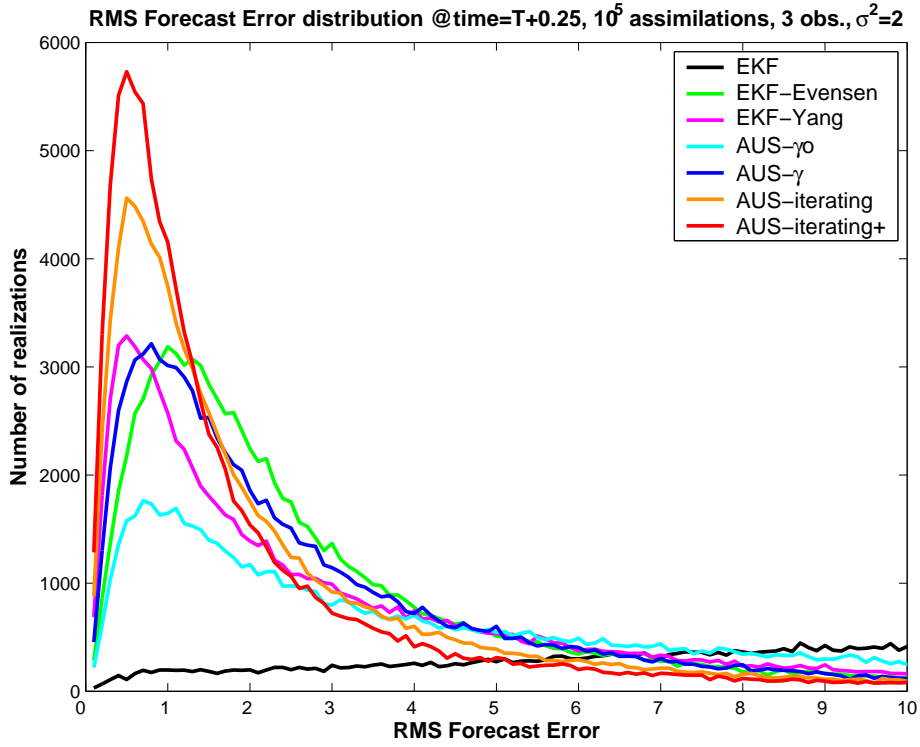


Figure 3.18: RMS Forecast Error distribution @ time= $T+0.25$ : an average on 100,000 assimilations with an assimilation window  $\tau = 0.25$ , 3 noisy observations with variance  $\sigma^2 = 2$

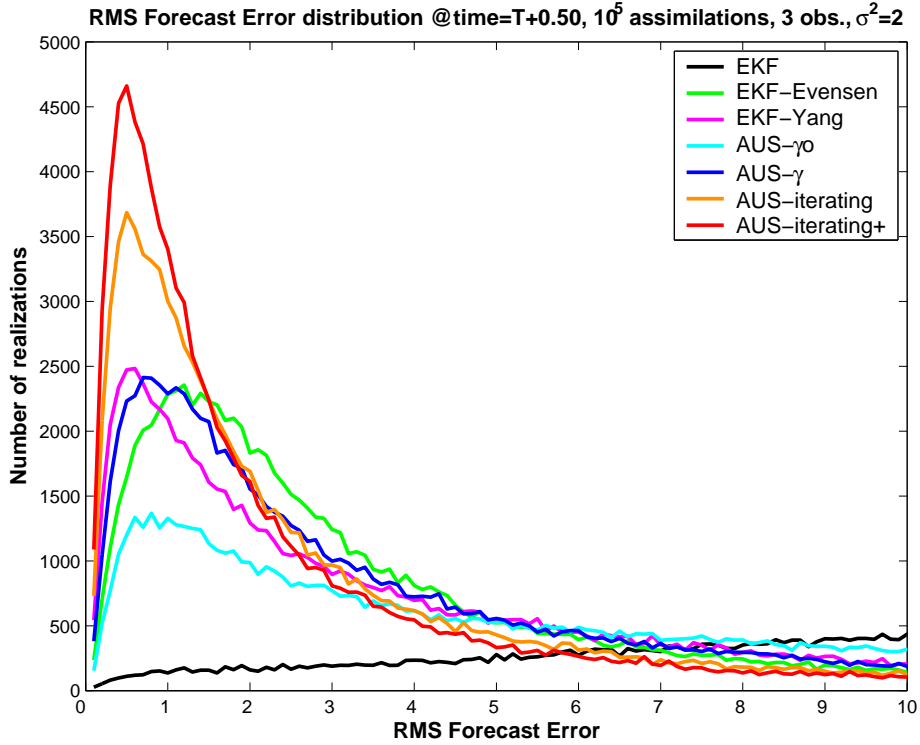


Figure 3.19: RMS Forecast Error distribution @ time= $T+0.50$ : an average on 100,000 assimilations with an assimilation window  $\tau = 0.25$ , 3 noisy observations with variance  $\sigma^2 = 2$

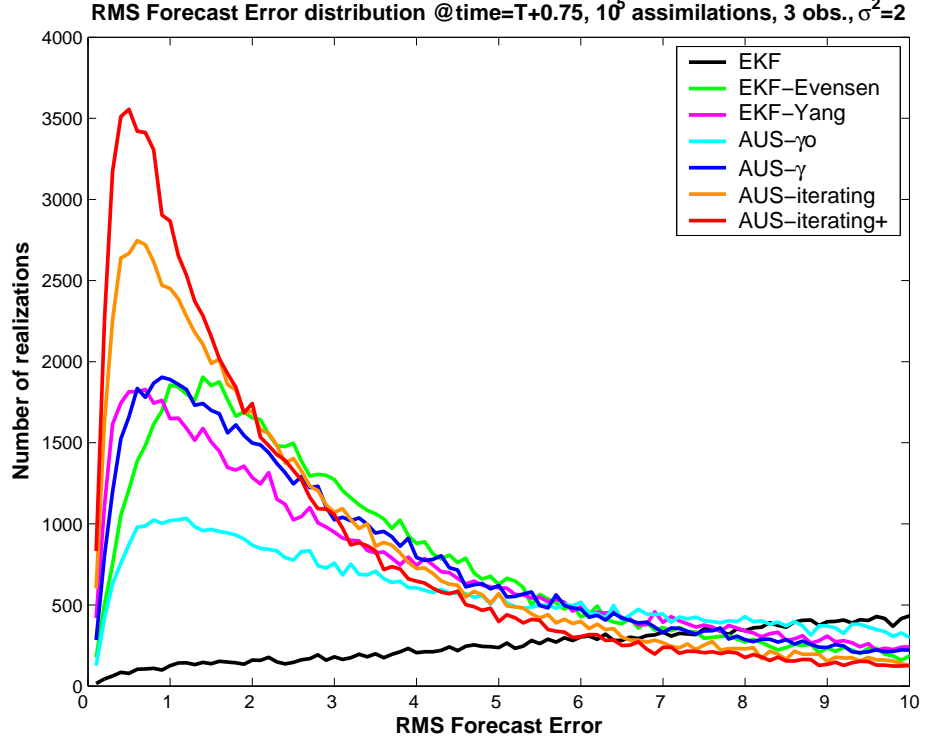


Figure 3.20: RMS Forecast Error distribution @ time= $T+0.75$ : an average on 100,000 assimilations with an assimilation window  $\tau = 0.25$ , 3 noisy observations with variance  $\sigma^2 = 2$

### 3.4.3 Noisy observations with variance $\sigma^2 = 1$

Now we're going to diminish the variance of the observational error. Since the relative performance are similar, we will only show the results for 3 observations probing. So the experimental context is:

- A 100,000 assimilations statistics
- 3 noisy observations at each assimilation step, with variance  $\sigma^2 = 1$
- an assimilation window  $\tau = 0.25$

The results, shown in Table 3.4 and in Figures 3.21-3.24, confirm those with  $\sigma^2 = 2$ . Note that EKF is still under-performing, and still experiencing frequent filter divergence.

### 3.4.4 Noisy observations with variance $\sigma^2 = 0.1$

Let's continue to lower observational noise variance: all the experiments are now performed under the following conditions:

- A 100,000 assimilations statistics

Table 3.4: RMS analysis and forecast error, an average over 100,000 assimilations. 3 noisy observations with variance  $\sigma^2 = 1$ . Assimilation window  $\tau = 0.25$ .

| <i>Assimilation technique</i> | <RMS analysis error> | <RMS forecast error> |          |          |
|-------------------------------|----------------------|----------------------|----------|----------|
|                               |                      | @ T+0.25             | @ T+0.50 | @ T+0.75 |
| EKF                           | 15.2                 | 16.3                 | 16.6     | 16.6     |
| EKF-Evensen                   | 1.30                 | 2.39                 | 4.94     | 5.93     |
| EKF-Yang                      | 0.99                 | 1.78                 | 3.04     | 3.63     |
| AUS- $\gamma_0$               | 3.55                 | 4.93                 | 6.65     | 7.50     |
| AUS- $\gamma$                 | 1.42                 | 2.52                 | 4.56     | 5.57     |
| AUS-iterating                 | 0.96                 | 1.94                 | 3.77     | 4.39     |
| AUS-iterating+                | 0.80                 | 1.63                 | 3.07     | 3.65     |

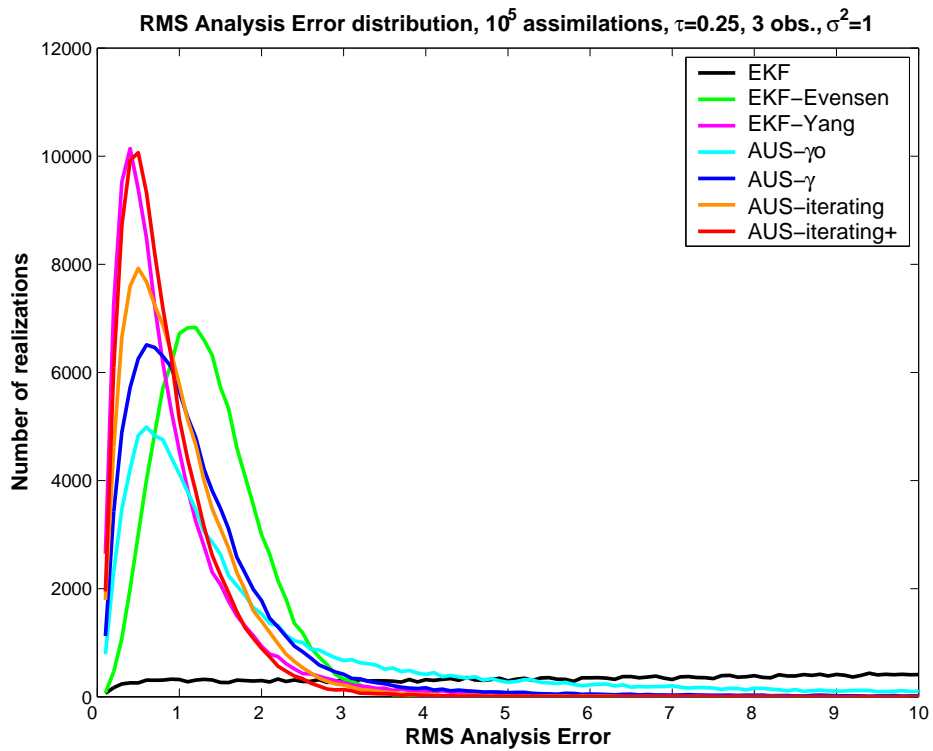


Figure 3.21: RMS Analysis Error distribution: an average on 100,000 assimilations with an assimilation window  $\tau = 0.25$ , 3 noisy observations with variance  $\sigma^2 = 1$

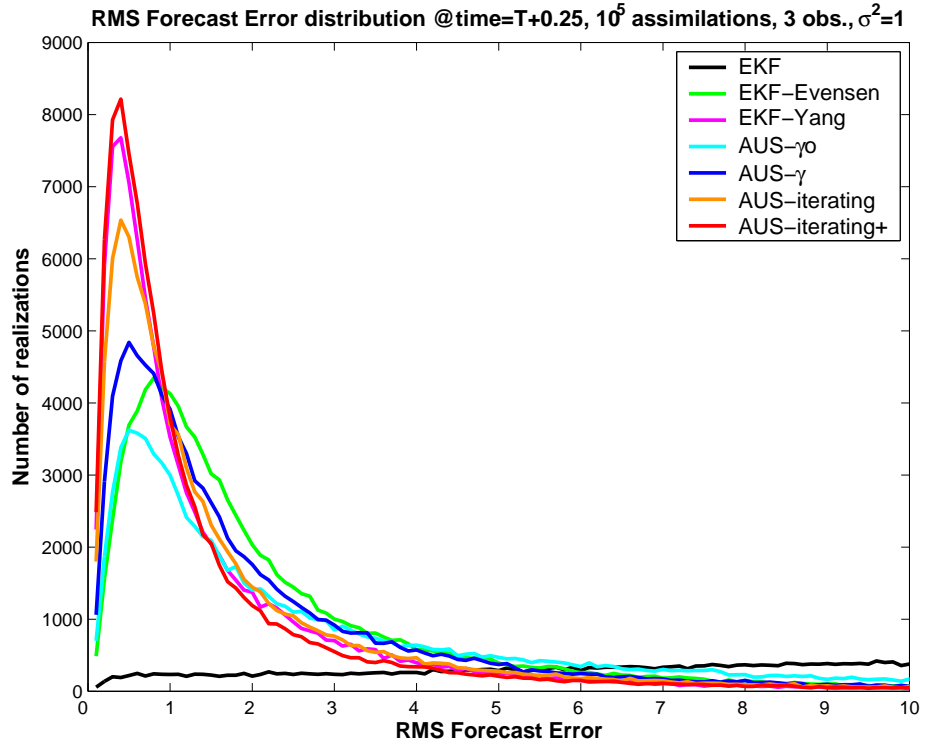


Figure 3.22: RMS Forecast Error distribution @ time= $T+0.25$ : an average on 100,000 assimilations with an assimilation window  $\tau = 0.25$ , 3 noisy observations with variance  $\sigma^2 = 1$

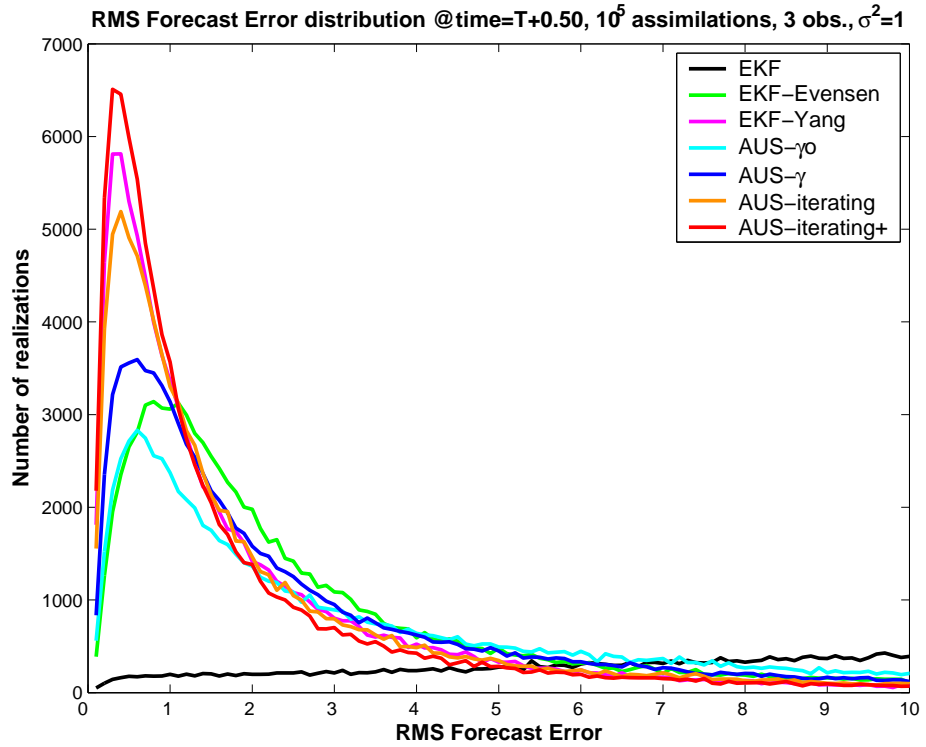


Figure 3.23: RMS Forecast Error distribution @ time= $T+0.50$ : an average on 100,000 assimilations with an assimilation window  $\tau = 0.25$ , 3 noisy observations with variance  $\sigma^2 = 1$

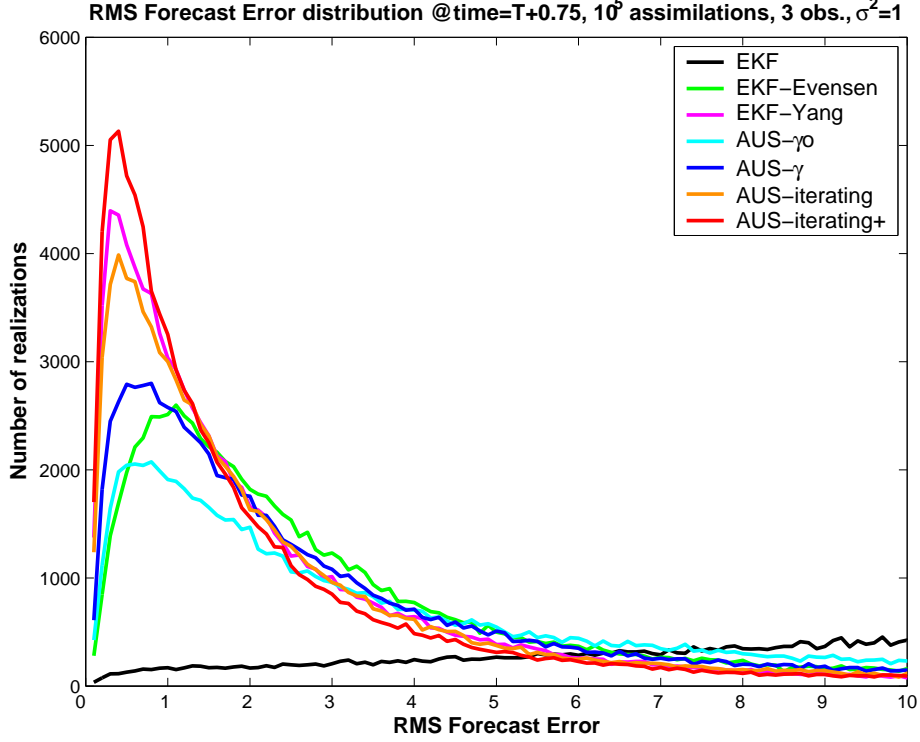


Figure 3.24: RMS Forecast Error distribution @ time= $T+0.75$ : an average on 100,000 assimilations with an assimilation window  $\tau = 0.25$ , 3 noisy observations with variance  $\sigma^2 = 1$

- 3 noisy observations each assimilation step, with variances  $\sigma^2 = 0.1 \Rightarrow \sigma = 0.316$
- an assimilation window  $\tau = 0.25$

The results are shown in Table 3.5 and in Figures 3.25-3.28, where all distributions are simply truncated where RMS error is equal to 5.

### 3.4.5 Root Mean Square forecast error: time dependence

Now we show the average time growth of RMS analysis error. The average is computed on 100000 assimilation steps. There are 3 variables observed, each assimilation window  $\tau = 0.25$ . At time  $t = 0$  the curves show the average RMS analysis error (see Tables 3.3, 3.4 and 3.5). The plots are listed on the basis of the variance of the observational error:

- In the 1st plot, Fig. 3.29:  $\sigma^2 = 2$
- in the 2nd plot, Fig. 3.30:  $\sigma^2 = 1$
- in the 3rd plot, Fig. 3.31:  $\sigma^2 = 0.1$

The EKF curve has been intentionally plotted only in the first one. Again, the AUS-iterating schemes, in particular AUS-iterating+, set the benchmark.

Table 3.5: RMS analysis and forecast error, an average over 100,000 assimilations. 3 noisy observations with variance  $\sigma^2 = 0.1 \Rightarrow \sigma = 0.316$ . Assimilation window  $\tau = 0.25$ .

| <i>Assimilation technique</i> | <RMS analysis error> | <RMS forecast error> |          |          |
|-------------------------------|----------------------|----------------------|----------|----------|
|                               |                      | @ T+0.25             | @ T+0.50 | @ T+0.75 |
| EKF                           | 15.0                 | 16.0                 | 16.4     | 16.4     |
| EKF-Evensen                   | 0.49                 | 0.79                 | 1.86     | 2.67     |
| EKF-Yang                      | 0.25                 | 0.53                 | 1.21     | 1.62     |
| AUS- $\gamma_0$               | 0.54                 | 1.02                 | 1.99     | 2.74     |
| AUS- $\gamma$                 | 0.36                 | 0.72                 | 1.56     | 2.20     |
| AUS-iterating                 | 0.30                 | 0.61                 | 1.37     | 1.85     |
| AUS-iterating+                | 0.26                 | 0.52                 | 1.14     | 1.53     |

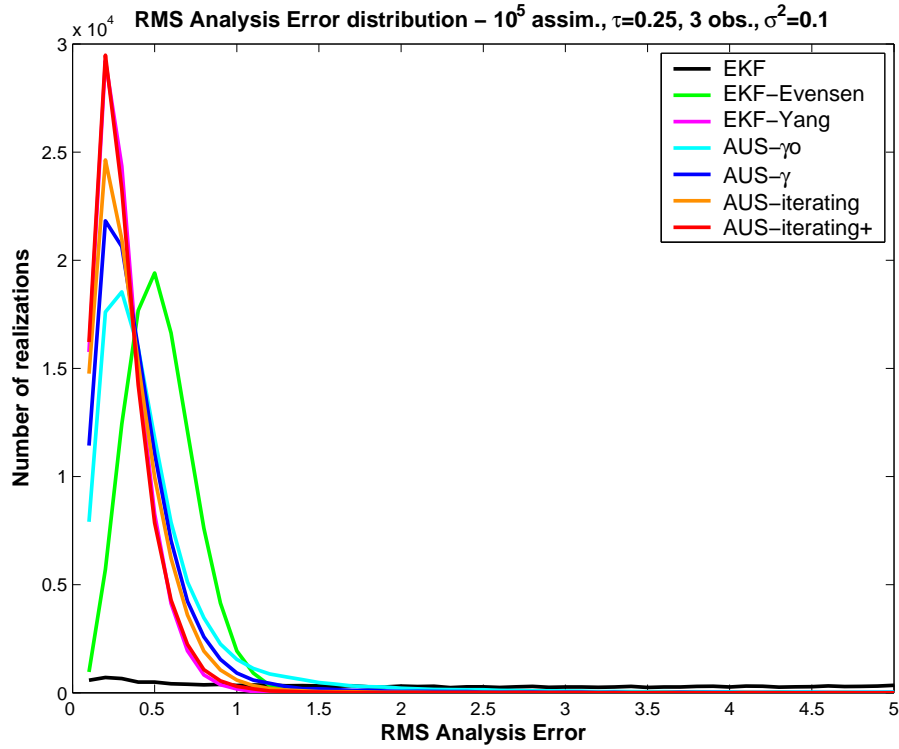


Figure 3.25: RMS Analysis Error distribution: an average on 100,000 assimilations with an assimilation window  $\tau = 0.25$ , 3 noisy observations with variance  $\sigma^2 = 0.1$

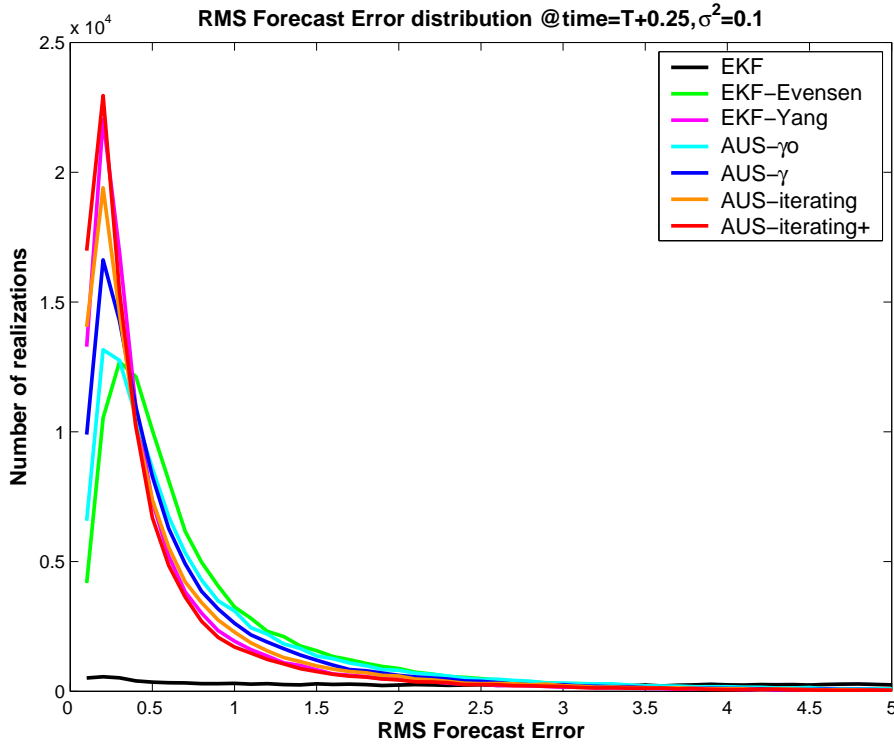


Figure 3.26: RMS Forecast Error distribution @ time= $T+0.25$ : an average on 100,000 assimilations with an assimilation window  $\tau = 0.25$ , 3 noisy observations with variance  $\sigma^2 = 0.1$

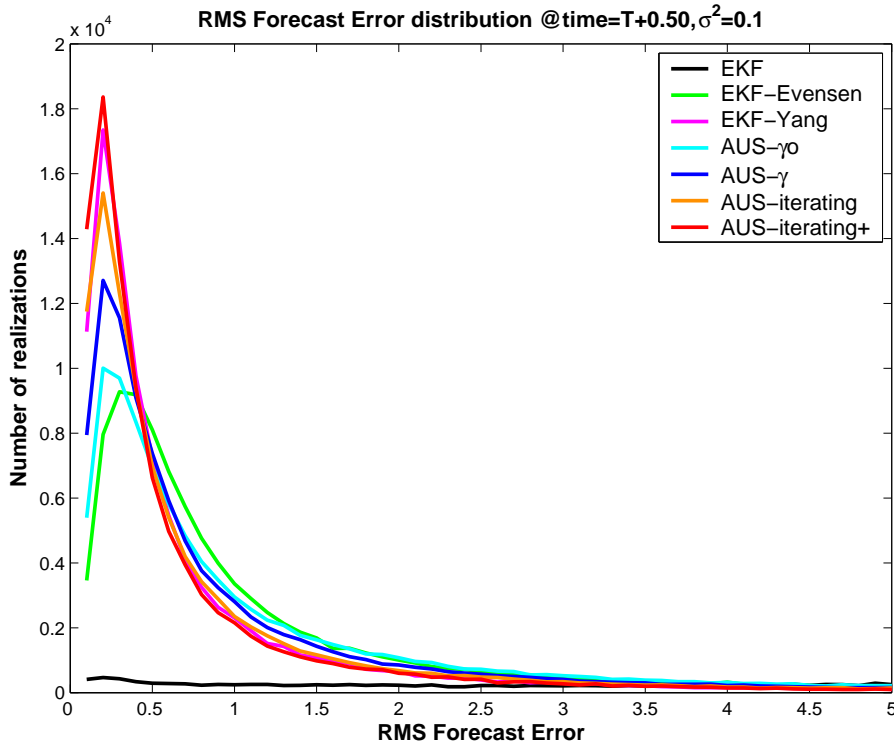


Figure 3.27: RMS Forecast Error distribution @ time= $T+0.50$ : an average on 100,000 assimilations with an assimilation window  $\tau = 0.25$ , 3 noisy observations with variance  $\sigma^2 = 0.1$



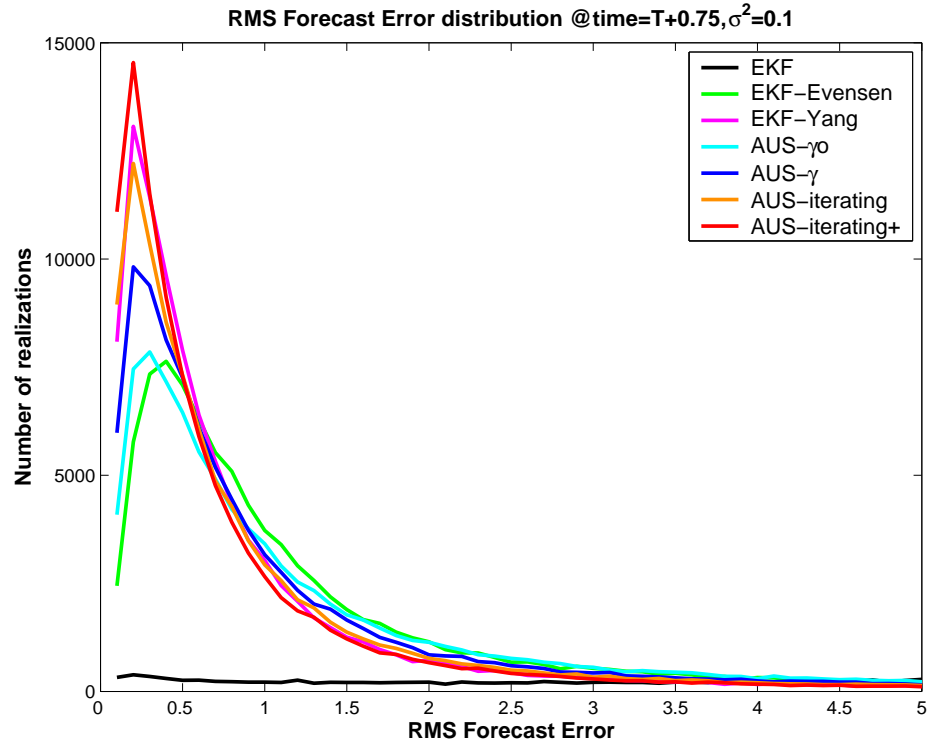


Figure 3.28: RMS Forecast Error distribution @ time= $T+0.75$ : an average on 100,000 assimilations with an assimilation window  $\tau = 0.25$ , 3 noisy observations with variance  $\sigma^2 = 0.1$

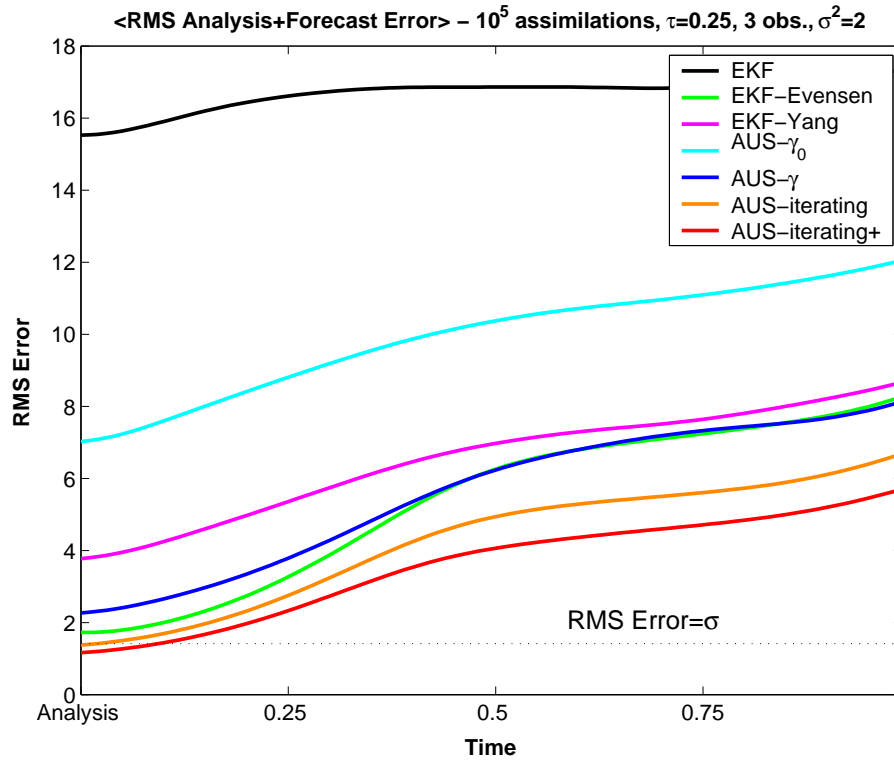


Figure 3.29: RMS analysis and forecast error: an average on 100,000 assimilation steps, assimilation window  $\tau = 0.25$ , 3 variables observed with variance  $\sigma^2 = 2 \Rightarrow \sigma = 1.414$ .

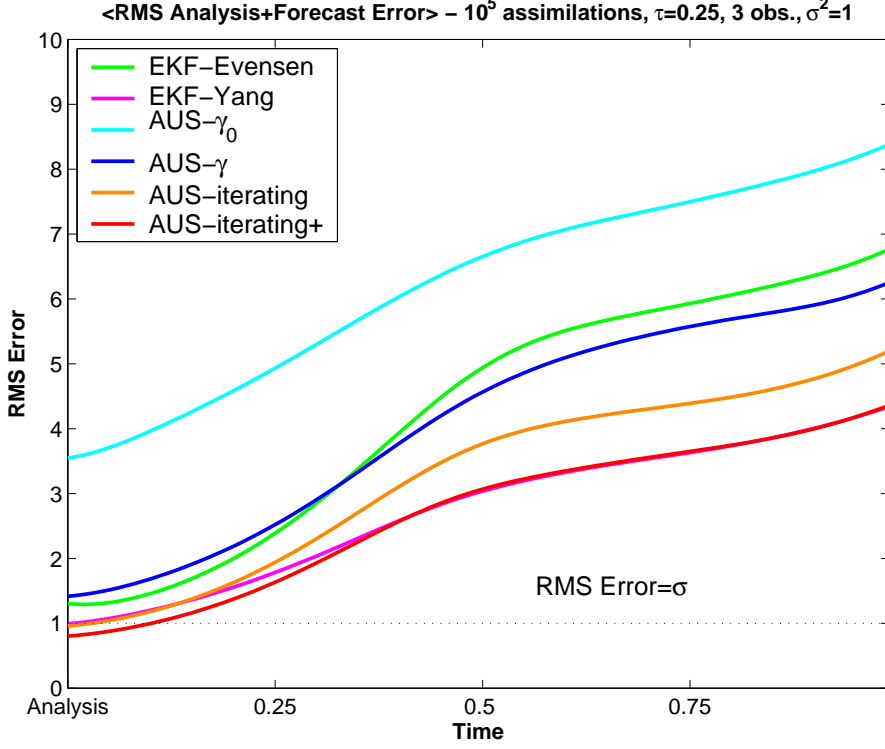


Figure 3.30: RMS analysis and forecast error: an average on 100,000 assimilation steps, assimilation window  $\tau = 0.25$ , 3 variables observed with variance 1.

### 3.5 Some illustrative examples

In this section we show some examples of noisy data assimilation, outputs of the different schemes probed in this work: in subsection 3.5.1 we present a case study with the same truth trajectory of the system (the variable  $x$  only), and the assimilation performances of our DA schemes. In subsection 3.5.2 we show a 3D plot of the AUS technique with some comments, while in subsection 3.5.3 we provide a step by step description AUS-iterating schemes.

#### 3.5.1 Comparing AUS with EKF assimilation schemes: case study

In Figures 3.32-3.38 we show, for the same case, the solution for the variable  $x$  and the analysis performed. The vertical green line shows the time of the last observation available for the assimilation. The experiment setup is:

- assimilation window  $\tau = 0.25$
- 3 variables observed with error variance  $\sigma^2 = 2$

We can see that, in this context, the EKF tends not to capture the regime changes, which is consistent with the average results over 100,000 assimilations shown in Table 3.3. EKF-Evensen

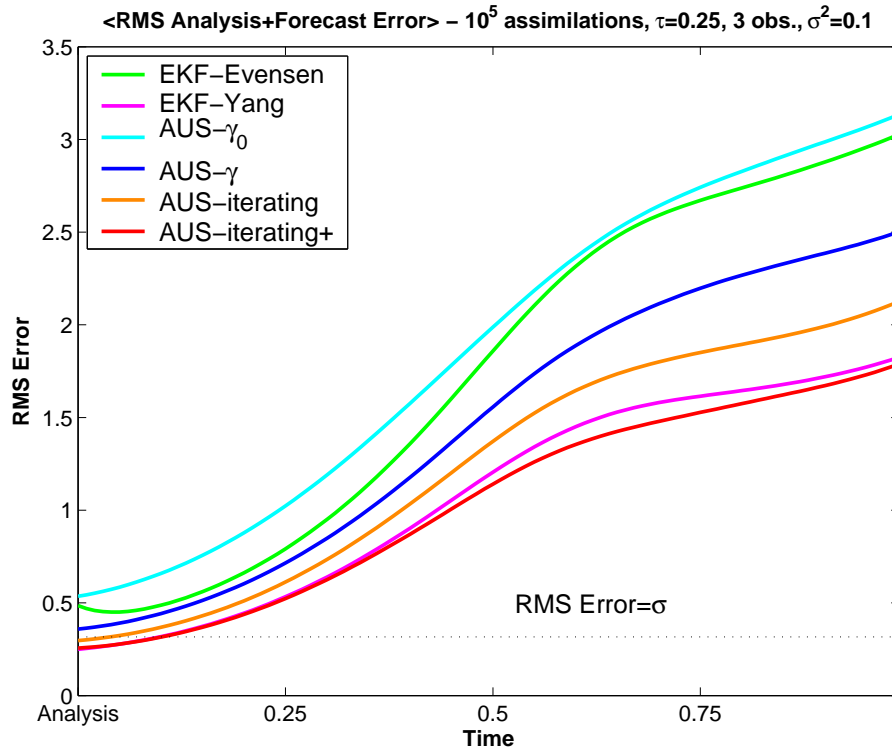


Figure 3.31: RMS analysis + forecast error: an average on 100,000 assimilation steps, assimilation window  $\tau = 0.25$ , 3 variables observed with variance  $\sigma^2 = 0.1 \Rightarrow \sigma = 0.316$ .

Figure 3.32: EKF assimilation scheme: solution for  $x$ . The green line shows the time of the last observation available.

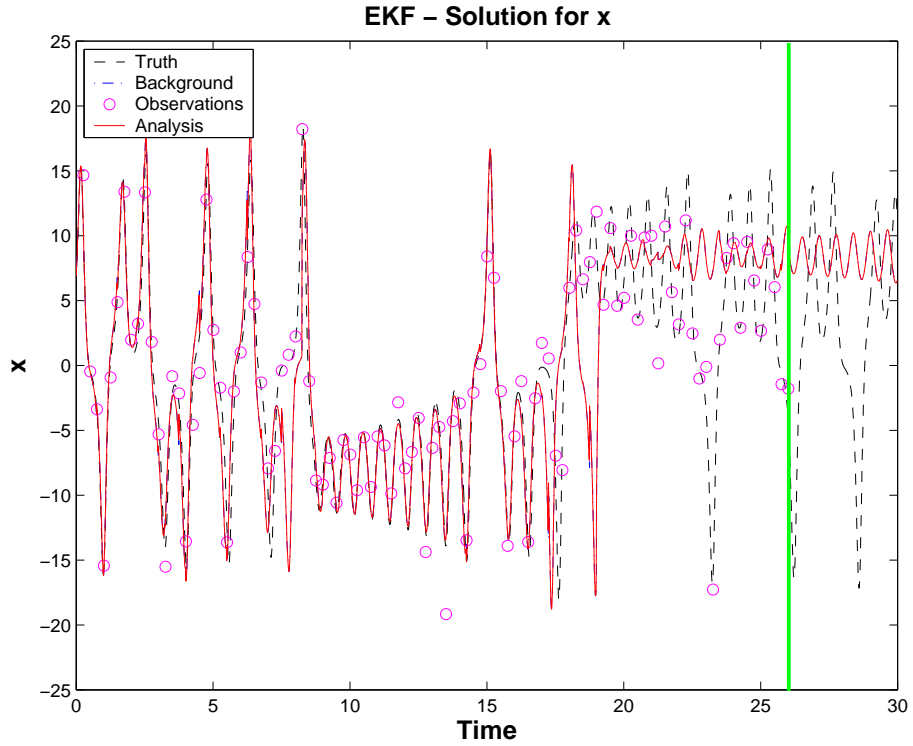


Figure 3.33: EKF-Evensen assimilation scheme: solution for  $x$ .

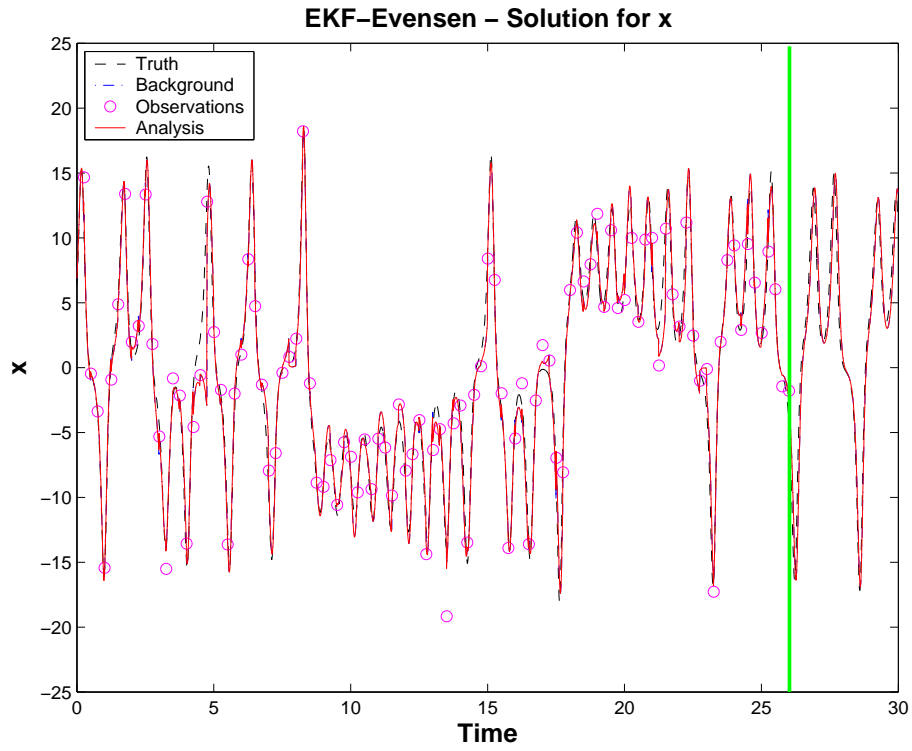


Figure 3.34: EKF-Yang assimilation scheme: solution for  $x$ .

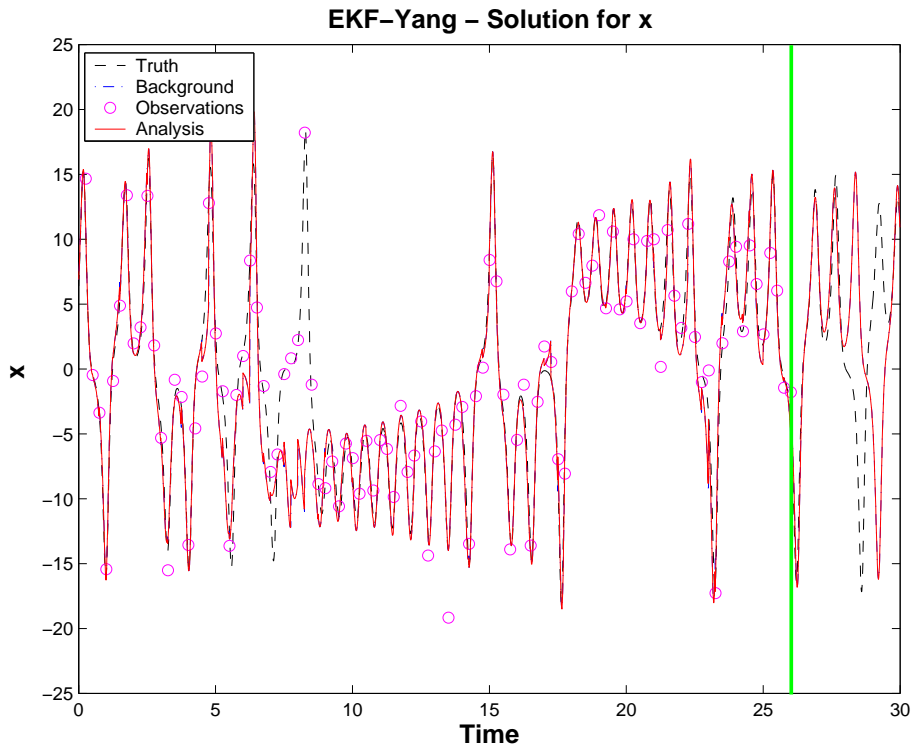


Figure 3.35: AUS- $\gamma_0$  assimilation scheme: solution for  $x$ .

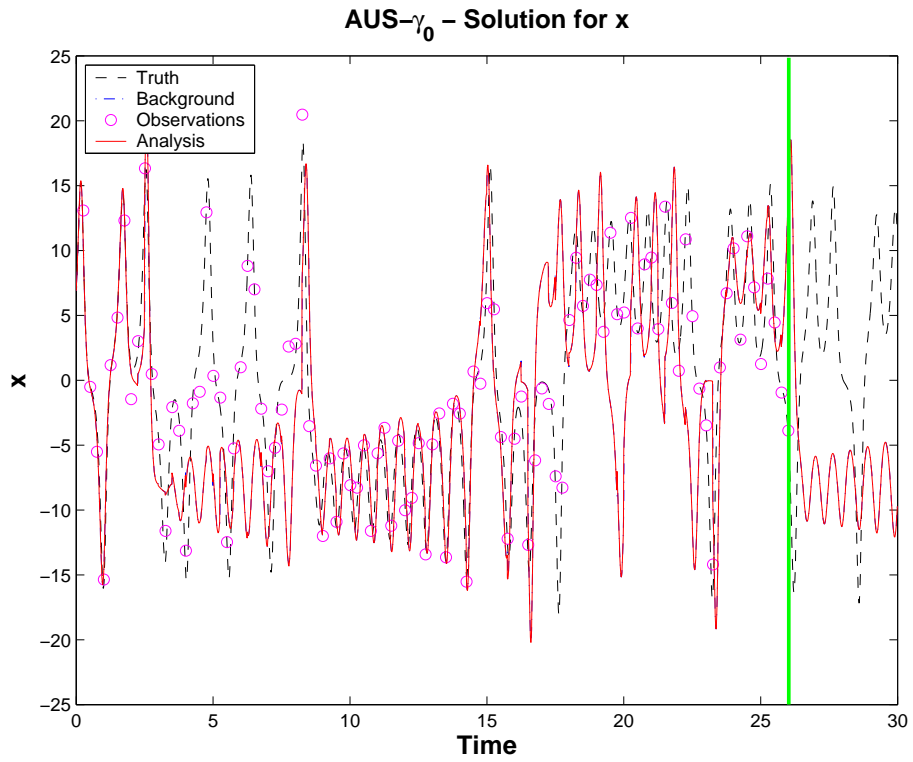


Figure 3.36: AUS- $\gamma$  assimilation scheme: solution for  $x$ .

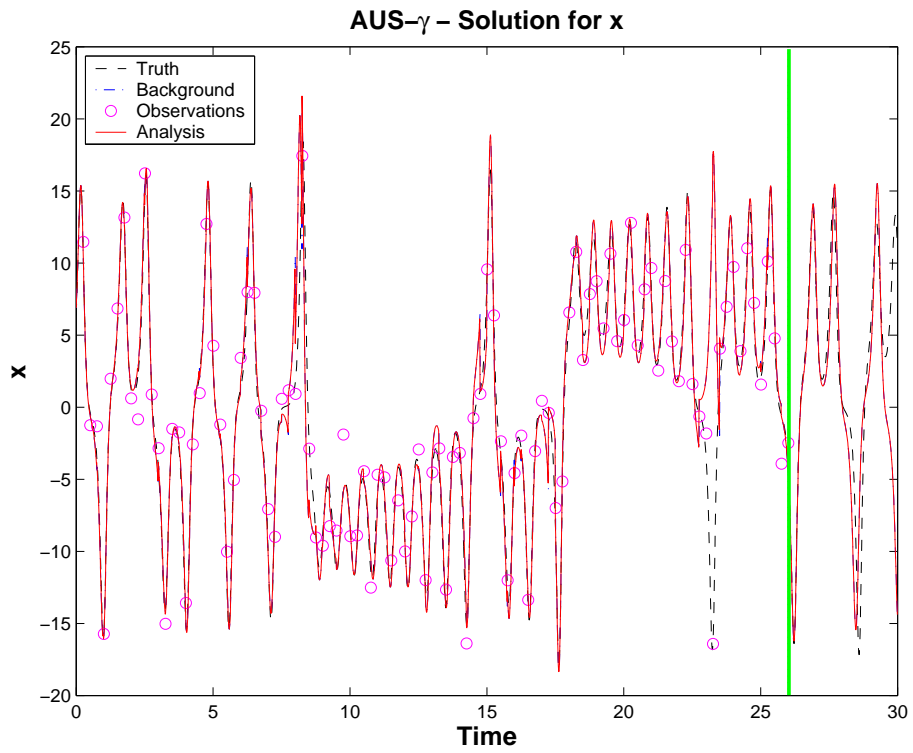


Figure 3.37: AUS-iterating assimilation scheme: solution for  $x$ .

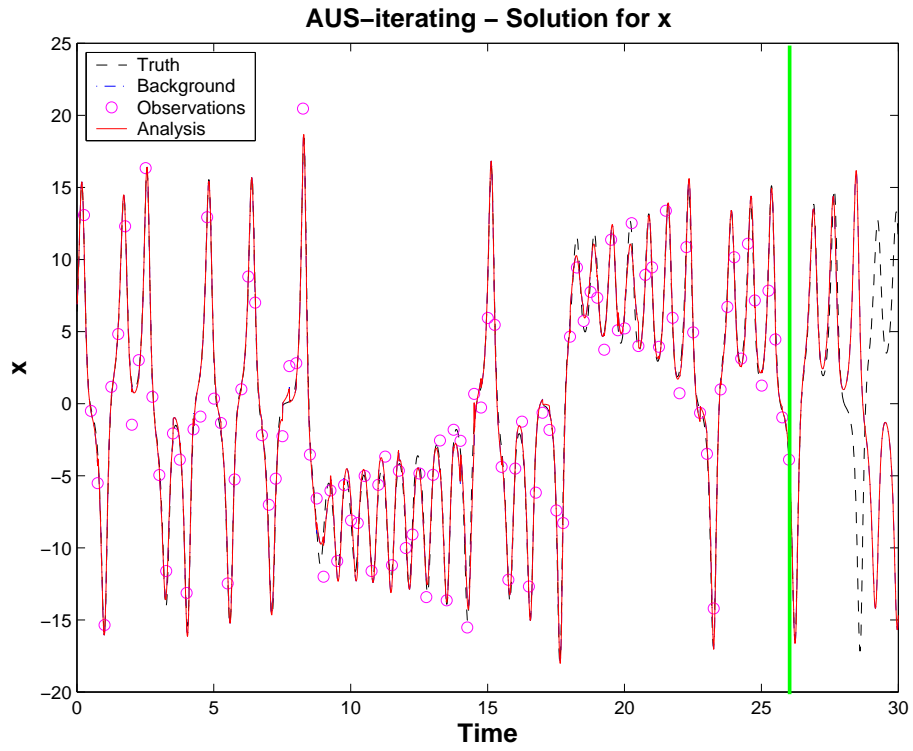


Figure 3.38: AUS-iterating+ assimilation scheme: solution for  $x$ .

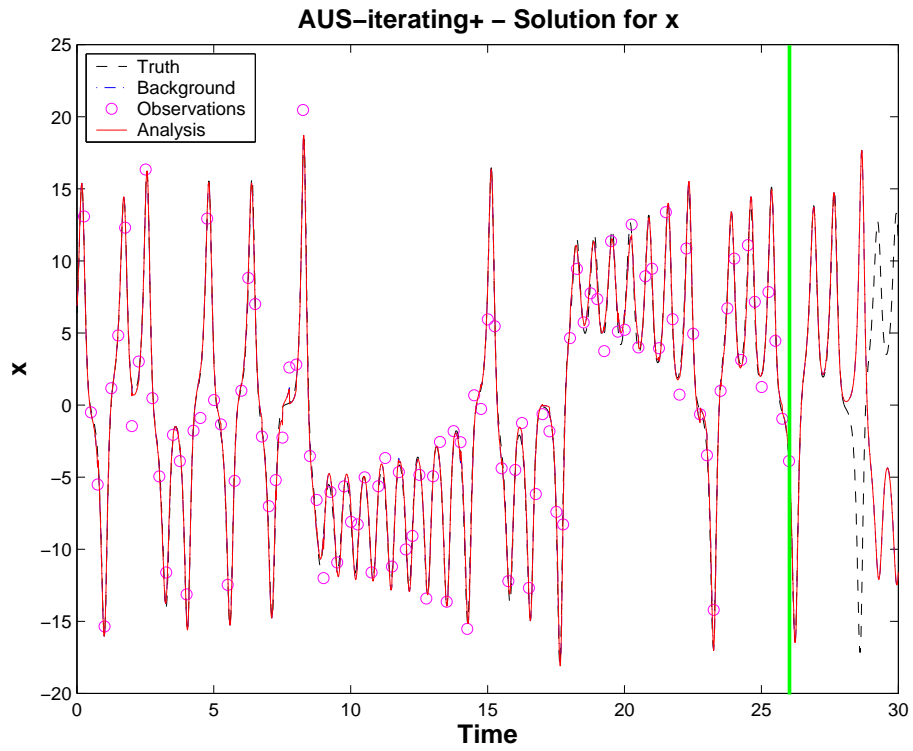
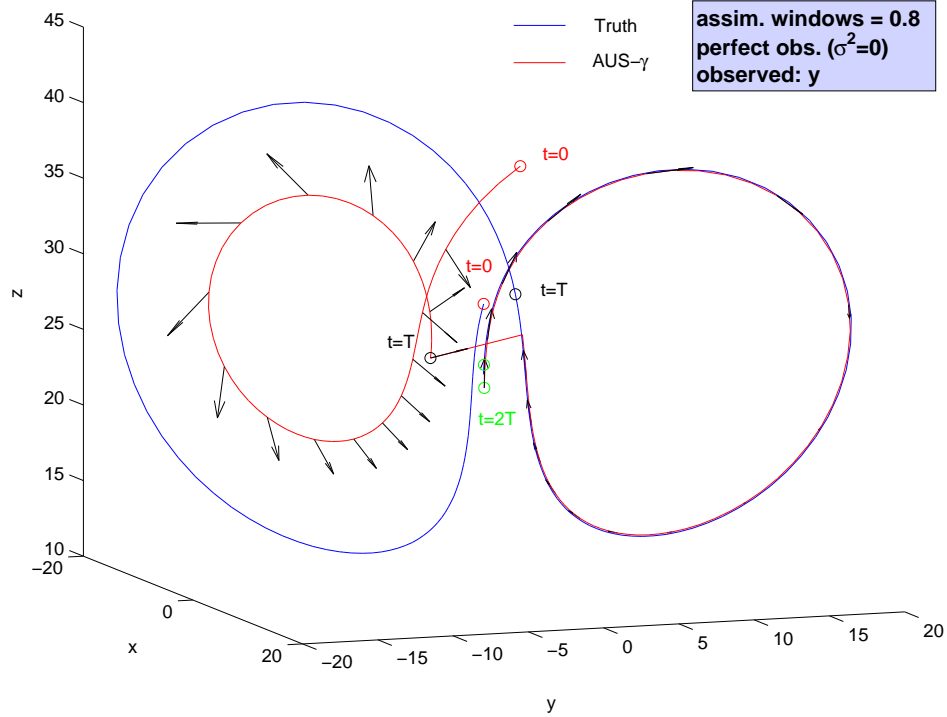


Figure 3.39: How AUS- $\gamma$  assimilates observations.



and EKF-Yang perform definitely better, particularly the former. As for the AUS schemes we can appreciate the improvements from the under-performing AUS- $\gamma_0$  up to AUS-iterating.

Note that EKF-Evensen has here a better performance in the forecast period than the AUS-iterating techniques, but this is not the general case: as already mentioned in Table 3.3, the average performance of EKF-Evensen is worse than that of AUS-iterating, both for analysis error and for forecast error.

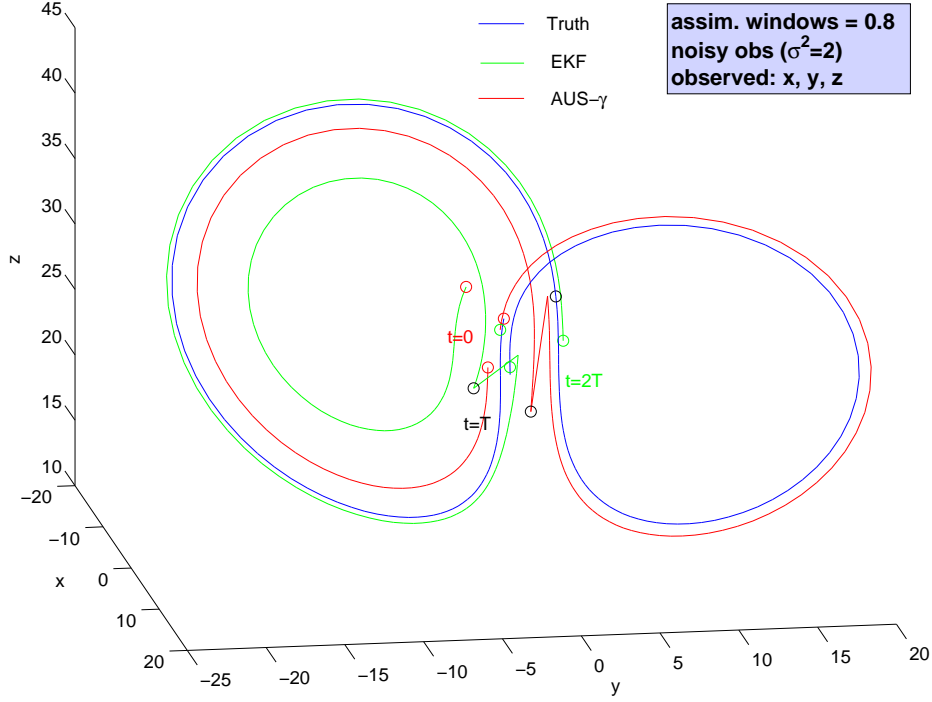
### 3.5.2 AUS- $\gamma$ : a 3D example

In Fig. 3.39 we show what is the essence of the AUS approach. We choose:

- a very long assimilation window,  $\tau = 0.8$
- only one observed variable:  $y$
- perfect observation,  $\sigma^2 = 0$

In the plot, the blue trajectory is the truth, the red one is the AUS- $\gamma$  assimilation trajectory. The red circles, labelled with  $t = 0$ , mark the beginning of both trajectories. The black circles ( $t = T$ ) show the same trajectories after one assimilation window, and the green circles ( $t = 2T$ ) one assimilation window ahead. The black vectors are the bred vectors estimated by BDAS

Figure 3.40: A qualitative comparison between EKF and AUS- $\gamma$ .



(see section 2.4) and evolved by the Tangent Linear Model operator, plotted every  $\Delta t = 0.05$ . When  $t = T$  the assimilation is done along the corresponding bred vector, assumed to capture the most unstable structure of the system (see section 2.3).

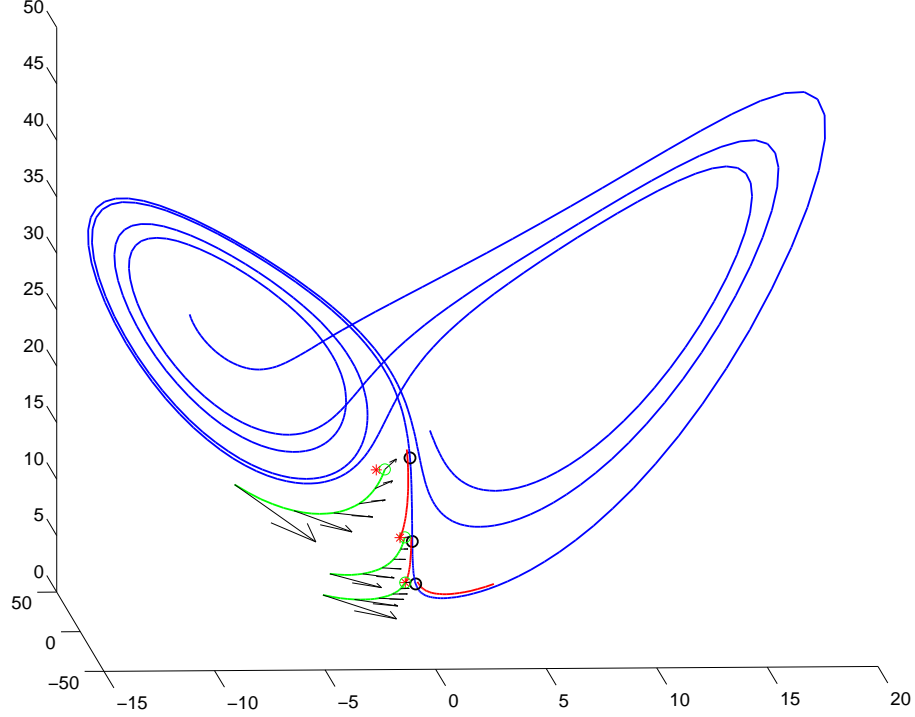
In Fig. 3.40 we show, as an example, the 3D comparison between EKF and AUS- $\gamma$ , in the following context:

- a very long assimilation window,  $\tau = 0.8$
- all 3 variables observed
- noisy observations, with  $\sigma^2 = 2$

Here the bred vectors for AUS- $\gamma$  have not been plotted for sake of clarity. The initial condition is the same for all schemes. After each DA technique does its job for 3 assimilation windows, we plot the trajectories starting from the the red circles ( $t = 0$ ). The blue trajectory is the truth, the red one is the AUS- $\gamma$  trajectory, while the green one is that of EKF. Both EKF and AUS- $\gamma$  are far from the truth, at the beginning, but at  $t = T$  (black circles) the EKF is not able to capture the regime change, while AUS- $\gamma$  does.



Figure 3.41: The zone of the attractor considered in the next Figure 3.42.



### 3.5.3 AUS-iterating: a step by step description

In Fig. 3.41 a 3D plot of the L63 system has been shown, with the highly unstable zone of the attractor, which is enlarged in Fig. 3.42. Here we set:

- a long assimilation window,  $\tau = 0.25$
- 2 adaptive observations
- noisy observations, with  $\sigma^2 = 2$

The blue trajectory is the truth. The green star is the forecast state, with which the first analysis can be computed (red circle); the red trajectory is the “first-attempt” forecast trajectory. Black vectors are the bred vector estimated by BDAS (see section 2.4) and evolved by the Tangent Linear Model operator, plotted every  $\Delta t = 0.05$ . When a new observation becomes available, a new assimilation can be done. In the iteration we go back to the previous assimilation step, correct the previous analysis according to equations 3.28-3.29 and to  $\Delta \mathbf{x}_k^a = c \exp(-\int_{t_k}^{t_{k+1}} \lambda(t) dt) \mathbf{e}_k$  (see subsection 3.3.4), and compute the final forecast trajectory (the green one). Of course new bred vectors are computed as well, but are shown only in Fig. 3.43 (in magenta).

Figure 3.42: How AUS-iterating works: a zoom of previous Fig. 3.41.

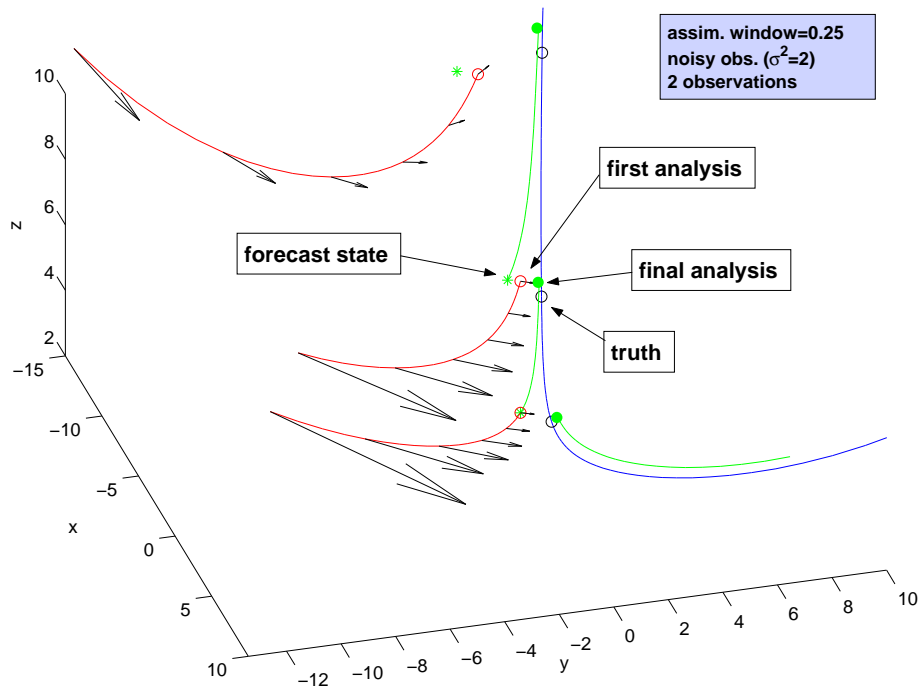
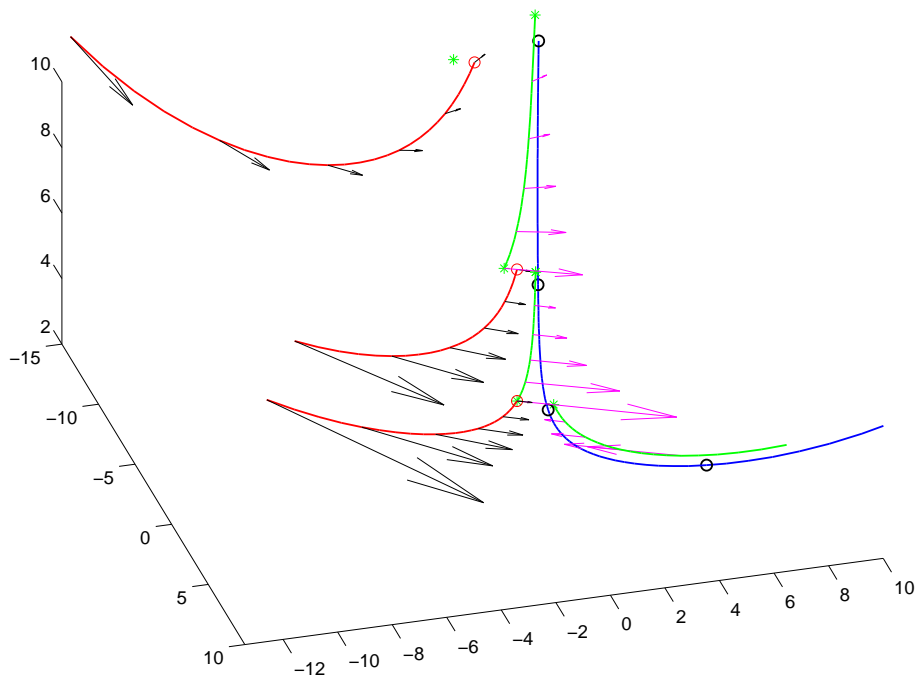


Figure 3.43: More details on AUS-iterating assimilation scheme, including the evolving unstable vectors in the final forecast trajectory, recomputed after iteration.



## 3.6 Adding two types of Model Error

We now wish to run experiments with some kind of model error in the equations of our L63 system. In subsection 3.6.1, we will start by simply adding a random term to each equation and at each integration step. In subsection 3.6.3, we will introduce a systematic error in our model by changing the value of the parameter  $r$ , which drives the convection instability (see subsection 1.4.2).

In short, we will compute the “true” trajectory of the system, we will get noisy observations of the true state of the system, but we will perform data assimilation using an imperfect model.

### 3.6.1 Random error

In these experiments we will simulate a systematic model error by adding, at each integration step  $dt = 0.01$ , a random term in the model equations (eq. 1.49):

$$\text{“Truth” System} \quad \left\{ \begin{array}{l} \frac{dx}{dt} = \sigma(y - x) \\ \frac{dy}{dt} = rx - y - xz \\ \frac{dz}{dt} = xy - bz \end{array} \right. \quad (3.42)$$

$$\text{Assimilation Model} \quad \left\{ \begin{array}{l} \frac{dx}{dt} = \sigma(y - x) + \epsilon_1 \\ \frac{dy}{dt} = rx - y - xz + \epsilon_2 \\ \frac{dz}{dt} = xy - bz + \epsilon_3 \end{array} \right. \quad (3.43)$$

where  $\epsilon_1, \epsilon_2, \epsilon_3$  are random additive terms, normally distributed with standard deviation  $A \times 0.1\sigma$ , with the parameter  $A$  assuming different values.

Table 3.6: Random error in the assimilation model: 3 observations with variance  $\sigma^2 = 2 \Rightarrow \sigma = 1.414$  each assimilation window  $\tau = 0.25$ . It is shown the mean RMS analysis error, an average over 20000 assimilations, for the different DA schemes.

| <b>3 obs</b>                  |         |         |         |
|-------------------------------|---------|---------|---------|
| <i>assimilation technique</i> | $A = 0$ | $A = 1$ | $A = 2$ |
| EKF                           | 15.6    | 15.5    | 15.7    |
| EKF-Evensen                   | 1.71    | 1.73    | 1.77    |
| EKF-Yang                      | 3.54    | 7.80    | 9.98    |
| AUS- $\gamma_0$               | 7.02    | 8.87    | 11.0    |
| AUS- $\gamma$                 | 2.24    | 2.72    | 3.89    |
| AUS-iterating                 | 1.35    | 1.94    | 2.75    |
| AUS-iterating+                | 1.15    | 1.80    | 2.44    |

### 3.6.2 Random error: comparing performances

All the experiments are performed under the following conditions:

- A 20,000 assimilations statistics
- 3 or 2 noisy observations with variance  $\sigma^2 = 2$
- an assimilation window  $\tau = 0.25$
- random additive terms  $\epsilon_{123}$  are added at each integration step  $dt = 0.01$ . They are normally distributed with standard deviation  $A \times 0.1\sigma$  ; the parameter  $A$  will assume the values  $A = 1$  and  $A = 2$
- in the 2 observation experiments, we will observe the two most convenient variables,  $y$  and  $z$ , for EKF schemes. For AUS schemes, we will observe the two variables associated with the two largest components of the unstable vector
- In EKF-Evensen scheme, the term  $\mathbf{Q}$  is the same as in eq. 3.8
- In EKF-Yang, no further tuning of the parameters has been done: the algorithm is the same as in experiments without model error

We show the results in Tables 3.6 and 3.7. The case with  $A = 0$ , i.e. without random error at all, is shown here for reference (even if it was already shown in paragraph 3.4 with a longer statistics).

The best AUS schemes are quite robust with respect to almost all the others. EKF-Evensen is actually more stable under these conditions, but it needs an estimate of the model error covariance matrix  $\mathbf{Q}$ .

Table 3.7: Random error in the assimilation model: 2 observations with variance  $\sigma^2 = 2 \Rightarrow \sigma = 1.414$  each assimilation window  $\tau = 0.25$ . We show the mean RMS analysis error, an average over 20000 assimilations.

| <b>2 obs</b>                  |         |         |         |
|-------------------------------|---------|---------|---------|
| <i>assimilation technique</i> | $A = 0$ | $A = 1$ | $A = 2$ |
| EKF                           | 15.5    | 15.6    | 15.6    |
| EKF-Evensen                   | 1.78    | 1.82    | 1.92    |
| EKF-Yang                      | 3.67    | 7.08    | 9.47    |
| AUS- $\gamma_0$               | 7.46    | 9.35    | 10.9    |
| AUS- $\gamma$                 | 2.50    | 3.04    | 4.19    |
| AUS-iterating                 | 1.54    | 2.22    | 3.27    |
| AUS-iterating+                | 1.28    | 2.11    | 3.07    |

### 3.6.3 Systematic error

In these experiments we will simulate a systematic error by augmenting the value of the parameter  $r$  in the model equations (eq. 1.49). As we have already seen in subsection 1.4.2, the parameter  $r$  is related to the intensity of the convection instability. We have to keep in mind that a value  $r = 28$  is a slightly supercritical value for unstable convection to occur. Thus the equations to be used are:

$$\text{"Truth" System} \quad \left\{ \begin{array}{l} \frac{dx}{dt} = \sigma(y - x) \\ \frac{dy}{dt} = rx - y - xz \\ \frac{dz}{dt} = xy - bz \end{array} \right. \quad (3.44)$$

$$\text{Assimilation Model} \quad \left\{ \begin{array}{l} \frac{dx}{dt} = \sigma(y - x) \\ \frac{dy}{dt} = (r + \Delta r)x - y - xz \\ \frac{dz}{dt} = xy - bz \end{array} \right. \quad (3.45)$$

The term  $\Delta r$  will introduce a systematic error in the equations used for data assimilation, by increasing the instability of convective motion in the model.

Table 3.8: Systematic error in the assimilation model,  $r = 28$  (i.e. no error, for reference),  $r = 30$ ,  $r = 33$ . Three observations with variance  $\sigma^2 = 2$  each assimilation window  $\tau = 0.25$ . It is shown the mean RMS analysis error, an average over 20,000 assimilations.

| <b>3 obs</b>                  |          |          |          |
|-------------------------------|----------|----------|----------|
| <i>assimilation technique</i> | $r = 28$ | $r = 30$ | $r = 33$ |
| EKF                           | 15.6     | 16.7     | 17.6     |
| EKF-Evensen                   | 1.71     | 1.74     | 1.86     |
| EKF-Yang                      | 3.54     | 11.4     | 13.9     |
| AUS- $\gamma_0$               | 7.02     | 11.5     | 14.6     |
| AUS- $\gamma$                 | 2.24     | 3.91     | 8.01     |
| AUS-iterating                 | 1.35     | 2.53     | 4.90     |
| AUS-iterating+                | 1.15     | 2.24     | 3.87     |

### 3.6.4 Systematic error: comparing performances

All the experiments are performed under the following conditions:

- A 20,000 assimilations statistics
- 3 or 2 noisy observations with variance  $\sigma^2 = 2$
- an assimilation window  $\tau = 0.25$
- systematic errors  $\Delta r = 0$  (for reference),  $\Delta r = 2$  and  $\Delta r = 5$
- in the 2 observation experiments we will observe the two most convenient variables,  $y$  and  $z$ , for EKF schemes; while for AUS schemes we will observe the two variables associated with the two largest components of the unstable vector
- In EKF-Evensen scheme, the term  $\mathbf{Q}$  is the same as in eq. 3.8
- In EKF-Yang, no further tuning of the parameters has been done: the algorithm is the same as in experiments without model error

We show the results in Tables 3.8 and 3.9. The case with  $r = 28$ , i.e. without systematic error, is included here for comparison (even if it was already shown in paragraph 3.4 with a longer statistics).

As for random error, all schemes considered turn out similar results: the EKF-Evensen scheme does outperform the others, but it needs an estimate of the model error covariance, which is not easy to do in realistic models. In these contexts, the best AUS schemes are still quite robust.

Table 3.9: Systematic error in the assimilation model,  $r = 28$  (i.e. no error, for reference),  $r = 30$ ,  $r = 33$ . Two observations with variance  $\sigma^2 = 2$  each assimilation window  $\tau = 0.25$ . It is shown the mean RMS analysis error, an average over 20000 assimilations.

**2 obs**

| <i>assimilation technique</i> | $r = 28$ | $r = 30$ | $r = 33$ |
|-------------------------------|----------|----------|----------|
| EKF                           | 15.5     | 16.9     | 17.9     |
| EKF-Evensen                   | 1.78     | 1.81     | 1.96     |
| EKF-Yang                      | 3.67     | 10.9     | 13.8     |
| AUS- $\gamma_0$               | 7.46     | 11.6     | 14.8     |
| AUS- $\gamma$                 | 2.50     | 4.26     | 8.21     |
| AUS-iterating                 | 1.54     | 2.76     | 5.38     |
| AUS-iterating+                | 1.28     | 2.53     | 4.88     |





## Chapter 4

# Conclusions

In this work we focused on the data assimilation problem for the highly nonlinear, chaotic Lorenz’s 63 system. Specifically, we tried to estimate the state of the system given a set of noisy observations at regular time intervals and the equations of the model.

In the contexts we have examined, including those with a model error, we have seen that Assimilation in the Unstable Subspace (AUS) once again has shown better efficiency than other advanced data assimilation schemes. Furthermore, at least for sufficiently long assimilation windows, our proposed approach for the estimate of the forecast error amplitude leads to a significant improvement of the assimilation performance. In the cases we have considered, the iterating extension of AUS improves over the standard AUS. In particular, it boosts the efficiency of regime’s changes tracking, with a low computational cost.

Other data assimilation schemes need estimates of *ad hoc* parameters, which have to be tuned for the specific model at hand. In NWP models, tuning of parameters — and in particular an estimate of the model error covariance matrix  $\mathbf{Q}$  — may turn out to be quite difficult. The estimate of  $\mathbf{Q}$ , on the other hand, is at the basis of the good performance of the Evensen model in particular in the presence of model error. As a final remark, we should note that the proposed approach may well be implemented in operational NWP models.



# Appendices



## Appendix A

# Euler and Runge-Kutta numerical integration methods

The Runge-Kutta methods are used to numerically approximate the solution of ordinary differential equations: they have much better performances with respect to the first order Euler method.

The kind of problems we want to solve is the following. Let's suppose we have a vector  $\mathbf{x}$  and a vector function  $\mathbf{f}(\mathbf{x})$  so that:

$$\frac{d}{dt}\mathbf{x} = \mathbf{f}(\mathbf{x}) \quad (\text{A.1})$$

Given the initial condition  $\mathbf{x}(t_0) = \mathbf{x}_0$ , we wish a systematic way to approximate the solution  $\mathbf{x}(t)$ .

### A.1 First order Euler method

The formula for the Euler method to advance a solution from  $t_n$  to  $t_{n+1} \equiv t_n + \Delta t$ , that is to estimate  $\mathbf{x}(t_n + \Delta t) \simeq \mathbf{x}_{n+1}$  given  $\mathbf{x}_n$ , comes straightforwardly from the definition of derivative:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{f}(\mathbf{x}_n) \Delta t + \mathcal{O}(\Delta t^2) \quad (\text{A.2})$$

where the error  $E = |\mathbf{x}(t_n + \Delta t) - \mathbf{x}_{n+1}|$  will tend to zero with  $\Delta t$ , i.e.  $E \propto \Delta t$ .

In the Euler method the solution  $\mathbf{x}_{n+1}$ , advanced through an interval  $\Delta t$ , is computed with the derivative information  $\mathbf{f}(\mathbf{x}_n)$  calculated at the initial time  $t_n$  only: it is not recommended for practical use because it is not so accurate, if compared to other methods, and it's not even very stable, too.

## A.2 RK2: second order Runge-Kutta scheme

A better convergence to zero of the error can be obtained with the 2nd order Runge-Kutta scheme, also known as *improved Euler method*, where the error  $E = |\mathbf{x}(t_n + \Delta t) - \mathbf{x}_{n+1}| \propto (\Delta t)^2$ .

Here the solution  $\mathbf{x}_{n+1}$  is computed by using an average for the derivative:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{1}{2}(\mathbf{k}_1 + \mathbf{k}_2) + \mathcal{O}(\Delta t^3) \quad (\text{A.3})$$

where

$$\mathbf{k}_1 = \mathbf{f}(\mathbf{x}_n) \Delta t \quad (\text{A.4})$$

$$\mathbf{k}_2 = \mathbf{f}(\mathbf{x}_n + \mathbf{k}_1) \Delta t = \mathbf{f}(\tilde{\mathbf{x}}_{n+1}) \Delta t \quad (\text{A.5})$$

The meaning of  $\mathbf{k}_1$ ,  $\mathbf{k}_2$  and  $\tilde{\mathbf{x}}_{n+1}$  is the following:

- $\mathbf{k}_1$  is a trial step to evaluate  $(\mathbf{x}_{n+1} - \mathbf{x}_n)$  through the Euler method, as in eq. A.2; the first evaluation of  $\mathbf{x}_{n+1}$  has been called  $\tilde{\mathbf{x}}_{n+1}$
- $\mathbf{k}_2$  is another estimate of  $(\mathbf{x}_{n+1} - \mathbf{x}_n)$  through the same Euler method, but using  $\mathbf{f}(\tilde{\mathbf{x}}_{n+1})$ , which is the derivative in the estimated end of the time interval

So RK2 (eq. A.3) exploits an average between the derivative  $\mathbf{f}(\mathbf{x}_n)$  calculated at the initial time  $t_n$  and the derivative  $\mathbf{f}(\tilde{\mathbf{x}}_{n+1}) = \mathbf{f}(\mathbf{x}_n + \mathbf{k}_1)$  which is an estimate of the derivative  $\mathbf{f}(\mathbf{x}_{n+1})$  calculated at the final time.

### A.3 RK4: fourth order Runge-Kutta scheme

We need not limit ourselves to the second order. A very popular Runge-Kutta method is the fourth order one, in which the error  $E = |\mathbf{x}(t_n + \Delta t) - \mathbf{x}_{n+1}| \propto (\Delta t)^4$ . Here we use a weighted average for the derivative:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{1}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4) + \mathcal{O}(\Delta t^5) \quad (\text{A.6})$$

where

$$\mathbf{k}_1 = \mathbf{f}(\mathbf{x}_n) \Delta t \quad (\text{A.7})$$

$$\mathbf{k}_2 = \mathbf{f}\left(\mathbf{x}_n + \frac{1}{2}\mathbf{k}_1\right) \Delta t \quad (\text{A.8})$$

$$\mathbf{k}_3 = \mathbf{f}\left(\mathbf{x}_n + \frac{1}{2}\mathbf{k}_2\right) \Delta t \quad (\text{A.9})$$

$$\mathbf{k}_4 = \mathbf{f}(\mathbf{x}_n + \mathbf{k}_3) \Delta t \quad (\text{A.10})$$

Higher order RK schemes are conceivable, but not necessarily better: they need more computational efforts.





## Appendix B

# Normalization factors for random variables

If  $x$  is a random variable with standard deviation  $\sigma$  and Probability Density Function (PDF):

$$P(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right) \quad (\text{B.1})$$

and  $\mathbf{x}$  is a vector whose components are random with standard deviation  $\sigma$  and the same PDF, it can be shown that:

$$\begin{aligned} \int_{-\infty}^{+\infty} |x| P(x) dx &= \sqrt{\frac{2}{\pi}} \sigma && \text{1-Dimensional case} \\ \int \int_{-\infty}^{+\infty} \|\mathbf{x}\| P(x_1) P(x_2) dx_1 dx_2 &= \sqrt{\frac{\pi}{2}} \sigma && \text{2-Dimensional case} \\ \int \int \int_{-\infty}^{+\infty} \|\mathbf{x}\| P(x_1) P(x_2) P(x_3) dx_1 dx_2 dx_3 &= \sqrt{\frac{8}{\pi}} \sigma && \text{3-Dimensional case} \end{aligned}$$



## Appendix C

# EKF, 1-Dimensional example

In the Extended Kalman Filter, if the system is described by a single variable only, the forecast and analysis covariance matrices and Kalman gain reduce to scalars.

### C.1 Forecast and analysis covariance matrices

From Extended Kalman Filter equations, we have:

$$K_k = \frac{p_k^f}{\sigma^2 + p_k^f} \quad (\text{C.1})$$

$$p_k^f = \alpha_k p_{k-1}^a \quad (\text{C.2})$$

$$p_k^a = (1 - K)p_k^f = \frac{\sigma^2 p_k^f}{\sigma^2 + p_k^f} \quad (\text{C.3})$$

Let's find forecast and analysis covariance matrices step by step:

$$\begin{aligned} p_0 &\equiv p_0^a \\ p_1^f &= \alpha_1 p_0 \\ p_1^a &= \frac{\sigma^2 p_1^f}{\sigma^2 + p_1^f} = \frac{\sigma^2 \alpha_1 p_0}{\sigma^2 + \alpha_1 p_0} \\ p_2^f &= \alpha_2 p_1^a = \frac{\sigma^2 p_0 \alpha_1 \alpha_2}{\sigma^2 + \alpha_1 p_0} \\ p_2^a &= \frac{\sigma^2 p_2^f}{\sigma^2 + p_2^f} = \frac{\sigma^2 p_0 \alpha_1 \alpha_2}{\sigma^2 + p_0(\alpha_1 + \alpha_1 \alpha_2)} \\ p_3^f &= \alpha_3 p_2^a = \frac{\sigma^2 p_0 \alpha_1 \alpha_2 \alpha_3}{\sigma^2 + p_0(\alpha_1 + \alpha_1 \alpha_2)} \\ p_3^a &= \frac{\sigma^2 p_3^f}{\sigma^2 + p_3^f} = \frac{\sigma^2 p_0 \alpha_1 \alpha_2 \alpha_3}{\sigma^2 + p_0(\alpha_1 + \alpha_1 \alpha_2 \alpha_3)} \end{aligned}$$

$$\begin{aligned}
& \vdots \quad \quad \quad \vdots \\
p_k^f &= \frac{\sigma^2 p_0 \alpha_1 \alpha_2 \dots \alpha_k}{\sigma^2 + p_0 (\alpha_1 + \alpha_1 \alpha_2 + \alpha_1 \alpha_2 \alpha_3 + \dots + \alpha_1 \alpha_2 \dots \alpha_{k-1})} = \frac{\sigma^2 p_0 \prod_{j=1}^k \alpha_j}{\sigma^2 + p_0 (\sum_{i=1}^{k-1} \prod_{j=1}^i \alpha_j)} \\
p_k^a &= \frac{\sigma^2 p_0 \alpha_1 \alpha_2 \dots \alpha_k}{\sigma^2 + p_0 (\alpha_1 + \alpha_1 \alpha_2 + \alpha_1 \alpha_2 \alpha_3 + \dots + \alpha_1 \alpha_2 \dots \alpha_k)} = \frac{\sigma^2 p_0 \prod_{j=1}^k \alpha_j}{\sigma^2 + p_0 (\sum_{i=1}^k \prod_{j=1}^i \alpha_j)} \\
K_k &= \frac{p_k^a}{\sigma^2}
\end{aligned}$$

In these equations we have also showed the result for Kalman gain  $K_k$ , that can be easily computed from  $K_k = \frac{P_k^f}{\sigma^2 + P_k^f}$ . The analysis, background and observational errors are bound by the following general equation [16]:

$$\epsilon^a = \epsilon^b + (\mathbf{B}^{-1} + \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{R}^{-1} (\epsilon^o - \mathbf{H} \epsilon^b) \quad (\text{C.4})$$

In this 1-dimensional example, if  $\mathbf{H} = \mathbf{1}$  and  $\mathbf{R} = \sigma^2$  it becomes:

$$\begin{aligned}
\epsilon^a &= \epsilon^b + \frac{\sigma^{-2}}{(p_k^f)^{-1} + \sigma^{-2}} (\epsilon^o - \mathbf{H} \epsilon^b) \\
&= \epsilon^b + \frac{p_k^f}{p_k^f + \sigma^2} (\epsilon^o - \epsilon^b) \\
&= (1 - K_k) \epsilon^b + K_k \epsilon^o
\end{aligned}$$

If  $\alpha_k = \alpha = \text{constant} > 1$ , then

$$\lim_{k \rightarrow \infty} p_k^f = \sigma^2 (\alpha - 1) \quad (\text{C.5})$$

$$\lim_{k \rightarrow \infty} p_k^a = \sigma^2 \frac{\alpha - 1}{\alpha} \quad (\text{C.6})$$

$$\lim_{k \rightarrow \infty} K_k = \frac{\alpha - 1}{\alpha} \quad (\text{C.7})$$

Indeed:

$$\begin{aligned}
\lim_{k \rightarrow \infty} p_k^f &= \lim_{k \rightarrow \infty} \frac{\sigma^2 p_0 \alpha^k}{\sigma^2 + p_0 (\alpha + \alpha^2 + \dots + \alpha^{k-1})} \\
&= \lim_{k \rightarrow \infty} \frac{\sigma^2 p_0 \alpha}{\frac{\sigma^2}{\alpha^{k-1}} + p_0 \left( \frac{1}{\alpha^{k-2}} + \dots + \frac{1}{\alpha} + 1 \right)} \\
&= \frac{\sigma^2 \alpha}{\left( \frac{1}{1 - \frac{1}{\alpha}} \right)} \\
&= \sigma^2 (\alpha - 1)
\end{aligned}$$

and

$$\begin{aligned}
\lim_{k \rightarrow \infty} p_k^a &= \lim_{k \rightarrow \infty} \frac{\sigma^2 p_0 \alpha^k}{\sigma^2 + p_0(\alpha + \alpha^2 + \dots + \alpha^k)} \\
&= \lim_{k \rightarrow \infty} \frac{\sigma^2 p_0}{\frac{\sigma^2}{\alpha^k} + p_0 \left( \frac{1}{\alpha^{k-1}} + \frac{1}{\alpha^{k-2}} + \dots + \frac{1}{\alpha} + 1 \right)} \\
&= \frac{\sigma^2}{\left( \frac{1}{1 - \frac{1}{\alpha}} \right)} \\
&= \sigma^2 \frac{\alpha - 1}{\alpha}
\end{aligned}$$

for if  $\alpha > 1$  then  $1/\alpha < 1$ . So:

$$\begin{aligned}
\lim_{k \rightarrow \infty} K_k &= \frac{\sigma^2(\alpha - 1)}{\sigma^2 + \sigma^2(\alpha - 1)} \\
&= \frac{\alpha - 1}{\alpha}
\end{aligned}$$

or, equivalently:

$$\lim_{k \rightarrow \infty} (1 - K_k) = \frac{1}{\alpha}$$

## C.2 Lyapunov exponents for free and forced systems

Now we redefine  $\alpha$  as the error growth (instead of the analysis error growth), so now  $1 - K \approx \frac{1}{\alpha^2}$ , or  $1 - K \approx \exp(-2\lambda\tau)$ . The error growth of the forced system will be:

$$e^{f\tau} = (1 - K)e^{\lambda\tau} \tag{C.8}$$

where  $f$  and  $\lambda$  are the greatest Lyapunov exponents of the forced and free systems respectively.

So, if  $\alpha_k = \alpha = \text{constant} > 1$ :

$$\begin{aligned}
e^{f\tau} &= (1 - K)e^{\lambda\tau} \\
&= \frac{1}{e^{2\lambda\tau}} e^{\lambda\tau} \\
&= e^{-\lambda\tau} \\
\Rightarrow f &= -\lambda
\end{aligned}$$

If  $\alpha_k$  are not constant the results are almost the same. If  $\alpha \gg 1$  then  $1 - K \approx 0$ , or  $K \approx 1$ ,

so the forecast will be almost ignored and the analysis error covariance matrix will tend to the observational error:  $p^a = \sigma^2(1 - 1/\alpha) \approx \sigma^2$ .

If we diminish the observational interval  $\tau$  ( $\tau \longrightarrow 0$ ):

$$p^a = \sigma^2(1 - \exp(-2\lambda\tau)) \longrightarrow 0 \quad (\text{C.9})$$

## Appendix D

# Inverse and pseudo-inverse matrices

An inverse matrix can be defined for square matrices only. The pseudo-inverse matrix, instead, is a generalization that can also be defined for rectangular matrices.

### D.1 Inverse matrix

Given an  $N \times N$  square matrix  $\mathbf{A}$ , the inverse matrix  $\mathbf{A}^{-1}$ , if it does exist, has the same dimensions as  $\mathbf{A}$  and is such that

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{A}\mathbf{A}^{-1} = \mathbf{I} \quad (\text{D.1})$$

where  $\mathbf{I}$  is the  $N \times N$  identity matrix. The matrix  $\mathbf{A}$  has the inverse matrix  $\mathbf{A}^{-1}$  if and only if  $\mathbf{A}$  is non-singular, i.e.:

$$\det(\mathbf{A}) = |\mathbf{A}| \neq 0 \quad (\text{D.2})$$

The matrix  $\mathbf{A}^{-1}$  can be written as follows:

$$\mathbf{A}^{-1} = \frac{1}{|\mathbf{A}|} \mathbf{C}^T \quad (\text{D.3})$$

where

$$\mathbf{C}^T = \begin{bmatrix} c_{11} & c_{21} & \cdots & c_{N1} \\ c_{12} & c_{22} & \cdots & c_{N2} \\ \vdots & \vdots & \ddots & \vdots \\ c_{1N} & c_{2N} & \cdots & c_{NN} \end{bmatrix} \quad (\text{D.4})$$

is the transpose of cofactor matrix of  $\mathbf{A}$ .

**Example.** Given a  $3 \times 3$  non-singular matrix  $\mathbf{A}$

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad (\text{D.5})$$

with  $\det(\mathbf{A}) = |\mathbf{A}| \neq 0$ , the cofactor matrix  $\mathbf{C}$  is

$$\mathbf{C} = \begin{bmatrix} +\det \begin{pmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{pmatrix} & -\det \begin{pmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{pmatrix} & +\det \begin{pmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{pmatrix} \\ -\det \begin{pmatrix} a_{12} & a_{13} \\ a_{32} & a_{33} \end{pmatrix} & +\det \begin{pmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{pmatrix} & -\det \begin{pmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{pmatrix} \\ +\det \begin{pmatrix} a_{12} & a_{13} \\ a_{22} & a_{23} \end{pmatrix} & -\det \begin{pmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{pmatrix} & +\det \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \end{bmatrix}$$

so its transpose is

$$\mathbf{C}^T = \begin{bmatrix} +(a_{22}a_{33} - a_{32}a_{23}) & -(a_{12}a_{33} - a_{32}a_{13}) & +(a_{12}a_{23} - a_{22}a_{13}) \\ -(a_{21}a_{33} - a_{31}a_{23}) & +(a_{11}a_{33} - a_{31}a_{13}) & -(a_{11}a_{23} - a_{21}a_{13}) \\ +(a_{21}a_{32} - a_{31}a_{22}) & -(a_{11}a_{32} - a_{31}a_{12}) & +(a_{11}a_{22} - a_{21}a_{12}) \end{bmatrix}$$



and the inverse matrix  $\mathbf{A}^{-1}$  is

$$\mathbf{A}^{-1} = \frac{1}{|\mathbf{A}|} \begin{bmatrix} +(a_{22}a_{33} - a_{32}a_{23}) & -(a_{12}a_{33} - a_{32}a_{13}) & +(a_{12}a_{23} - a_{22}a_{13}) \\ -(a_{21}a_{33} - a_{31}a_{23}) & +(a_{11}a_{33} - a_{31}a_{13}) & -(a_{11}a_{23} - a_{21}a_{13}) \\ +(a_{21}a_{32} - a_{31}a_{22}) & -(a_{11}a_{32} - a_{31}a_{12}) & +(a_{11}a_{22} - a_{21}a_{12}) \end{bmatrix}$$

Lastly, we recall some other useful facts: it can be shown that if the matrix  $\mathbf{A}$  can be written as a product, then the inverse matrix  $\mathbf{A}^{-1}$  can be expressed in term of the product of the inverse matrices:

$$\mathbf{A} = \mathbf{B}\mathbf{C} \implies \mathbf{A}^{-1} = \mathbf{C}^{-1}\mathbf{B}^{-1} \quad (\text{D.6})$$

If the matrix  $\mathbf{A}$  admits the inverse  $\mathbf{A}^{-1}$  then we have:

$$(\mathbf{A}^T)^{-1} = (\mathbf{A}^{-1})^T \quad (\text{D.7})$$

$$(k\mathbf{A})^{-1} = k^{-1}\mathbf{A}^{-1} \quad (\text{D.8})$$

## D.2 Pseudo-inverse matrix

Given an  $M \times N$  matrix  $\mathbf{A}$ , the pseudo-inverse  $\mathbf{A}^+$  has the same dimension as  $\mathbf{A}^T$ , and satisfies the following relations:

$$\mathbf{A}\mathbf{A}^+\mathbf{A} = \mathbf{A} \quad (\text{D.9})$$

$$\mathbf{A}^+\mathbf{A}\mathbf{A}^+ = \mathbf{A}^+ \quad (\text{D.10})$$

$$(\mathbf{A}\mathbf{A}^+)^* = \mathbf{A}\mathbf{A}^+ \implies \mathbf{A}\mathbf{A}^+ \text{ is hermitian} \quad (\text{D.11})$$

$$(\mathbf{A}^+\mathbf{A})^* = \mathbf{A}^+\mathbf{A} \implies \mathbf{A}^+\mathbf{A} \text{ is hermitian} \quad (\text{D.12})$$

where  $\mathbf{A}^*$  is the conjugate transpose of the matrix  $\mathbf{A}$ , i.e. the matrix computed by taking the transpose of  $\mathbf{A}$  and then the complex conjugate of each element. If the matrix  $\mathbf{A}$  is real, then  $\mathbf{A}^* = \mathbf{A}^T$ . Should exist the inverse of  $\mathbf{A}^*\mathbf{A}$ , then

$$\mathbf{A}^+ = (\mathbf{A}^*\mathbf{A})^{-1}\mathbf{A}^T \quad (\text{D.13})$$

If it does exist the inverse  $\mathbf{A}^{-1}$  of the matrix  $\mathbf{A}$ , then it can be shown that  $\mathbf{A}^{-1} = \mathbf{A}^+$ .

**Example.** Given a  $3 \times 2$  real matrix  $\mathbf{A}$

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 2 & 5 \\ 7 & 3 \end{bmatrix}$$

then its pseudo-inverse  $\mathbf{A}^+$  is

$$\mathbf{A}^+ = \begin{bmatrix} 0.0389 & -0.0994 & 0.1657 \\ -0.0354 & 0.2377 & -0.0629 \end{bmatrix}$$

which satisfies the four properties above:

$$(\mathbf{A}\mathbf{A}^+)^T = \mathbf{A}\mathbf{A}^+ = \begin{bmatrix} 0.0389 & -0.0994 & 0.1657 \\ -0.0994 & 0.9897 & 0.0171 \\ 0.1657 & 0.0171 & 0.9714 \end{bmatrix}$$

$$(\mathbf{A}^+\mathbf{A})^T = \mathbf{A}^+\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Furthermore, since

$$\mathbf{A}^T\mathbf{A} = \begin{bmatrix} 54 & 31 \\ 31 & 34 \end{bmatrix}$$

then

$$(\mathbf{A}^T\mathbf{A})^{-1} = \begin{bmatrix} 0.0389 & -0.0354 \\ -0.0354 & 0.0617 \end{bmatrix}$$

and

$$\mathbf{A}^+ = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T = \begin{bmatrix} 0.0389 & -0.0994 & 0.1657 \\ -0.0354 & 0.2377 & -0.0629 \end{bmatrix}$$

# Bibliography

- [1] Andrews, D. G., 2000: *An Introduction to Atmospheric Physics*. Cambridge University Press. Cambridge, UK.
- [2] Arnold, V. I., 1986: *Metodi Matematici della Meccanica Classica*. Editori Riuniti Edizioni MIR, Roma, Italia.
- [3] Carrassi, A., Trevisan, A. and Uboldi, F. 2006: Adaptive observations and assimilation in the unstable subspace by breeding on the data assimilation system. *Tellus*, **59A-1**, 101-113.
- [4] Charney, J. G., 1951: Dynamical forecasting by numerical process. *Compendium of meteorology*. American Meteorological Society, Boston, MA, USA.
- [5] Chin, T. M., Turmon, M. J., Jewell, J. B., Ghil, M. 2005. An ensemble-based smoother with optimized weights for highly nonlinear systems, in print.
- [6] Corazza, M., Kalnay, E., Patil, D. J., Yang, S. C., Morss, R., Cai, M., Szunyogh, I., Hunt, B. R. and Yorke, J. A. 2003: Use of the breeding technique to estimate the structure of the analysis “error of the day”. *Nonlin. Processes Geophys.* **10**, 223-243.
- [7] Courtier, P. and O. Talagrand, 1990: Variational assimilation of meteorological observations with the direct and adjoint shallow water equations. *Tellus*, **42A**, 531-549.
- [8] Daley, R., 1991: *Atmospheric Data Analysis*. Cambridge University Press. Cambridge, UK.
- [9] Ehrendorfer, M., 2002: The Liouville equation in atmospheric predictability. *ECMWF Seminar Proceedings series 2002, Predictability of Weather and Climate*.
- [10] Evensen, G., 1994: Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *J. Geophys. Res.* **99** (C5), 10143-10162.
- [11] Evensen, G., 1997: Advanced data assimilation for strongly nonlinear dynamics. *Mon. Wea. Rev.*, **125**, 1342-1354.

- [12] Hénon, M., 1976: A two-dimensional mapping with a strange attractor. *Comm. Math. Phys.* **50**, 69-77.
- [13] Holton, J. R., *An Introduction to Dynamic Meteorology*. Academic Press. Orlando, USA.
- [14] Ide, K., P. Courtier, M. Ghil and A. C. Lorenc, 1997: Unified notation for data assimilation: Operational, sequential and variational. *J. Meteor. Soc. Japan*, **75**, 181-189.
- [15] Jazwinski, A. H., 1970: *Stochastic Processes and Filtering Theory*. Academic Press, New York, NY, USA.
- [16] Kalnay, E., 2003: *Atmospheric Modeling, Data Assimilation and Predictability*. Cambridge University Press. Cambridge, UK.
- [17] Lewis, J. and J. Derber, 1985: The use of adjoint equations to solve a variational adjustment problem with advective constraint. *Tellus*, **37A**, 309-322.
- [18] Lorenz, E. N., 1963: Deterministic non-periodic flows. *J. Atmos. Sci.* **20**, 130-141.
- [19] Lorenz, E. N., 1993: *The Essence of Chaos*. University of Washington Press. Seattle, USA.
- [20] Lorenz, E. N., 1996: Predictability: A problem partly solved. *Proc. Seminar on Predictability*, Vol. 1, Reading, United Kingdom, ECMWF, 1-18.
- [21] Lorenz, E. N. and K. A. Emanuel, 1998: Optimal sites for supplementary weather observations: simulation with a small model. *J. Atmos. Sci.* **55**, 399-414.
- [22] May, R. M., 1974: *Stability and Complexity in Model Ecosystems*. Princeton University Press. Princeton, NJ, USA.
- [23] Maybeck, P. S., 1979: *Stochastic Models: Estimation and Control, Volume 1*. Academic Press, New York, NY, USA.
- [24] Nicolis, C., 2003: Dynamics of Model Error: Some Generic Features. *J. Atmos. Sci.* **60**, 2208-2218.
- [25] Nicolis, C., 2004: Dynamics of Model Error: The Role of Unresolved Scales Revisited. *J. Atmos. Sci.* **61**, 1740-1753.
- [26] Ott E., Sauer T. and Yorke J. A., 1994: *Coping with Chaos: Analysis of Chaotic Data and the Exploitation of Chaotic Systems*. John Wiley & Sons Inc., New York.
- [27] Peixoto, J. P. and A. H. Oort, 1992: *Physics of Climate*. American Institute of Physics, New York, USA.

- [28] Poincaré, J. H., 1997 (reprint): *Scienza e Metodo*. Giulio Einaudi editore SpA, Torino, Italia.
- [29] Pratt, W. J., Raiffa H. and Schlaifer R., 1995: *Introduction to Statistical Decision Theory*. The MIT Press, Cambridge, MA, USA.
- [30] Saltzman, B., 1962: Finite amplitude free convection as an initial value problem—I. *J. Atmos. Sci.* **19**, 329-341.
- [31] Strogatz, S. H, 1994: *Nonlinear Dynamics and Chaos*. Westview Press. Cambridge, USA.
- [32] Trevisan, A. and F. Pancotti, 1998: Periodic orbits, Lyapunov vectors and singular vectors in the Lorenz system. *J. Atmos. Sci.* **55**, 390-398.
- [33] Trevisan, A. and F. Uboldi, 2004: Assimilation of Standard and Targeted Observations within the Unstable Subspace of the Observation-Analysis-Forecast Cycle System. *J. Atmos. Sci.* **61**, 103-113.
- [34] Uboldi, F., Trevisan, A. and Carrassi, A. 2005: Developing a dynamically based assimilation method for targeted and standard observations. *Nonlin. Processes in Geophys.* **12**, 149-156.
- [35] Uboldi, F. and A. Trevisan, 2006: Detecting unstable structures and controlling error growth by assimilation of standard and adaptive observations in a primitive equation ocean model. *Nonlin. Processes in Geophys.* **13**, 67-81.
- [36] Yang *et al.*, 2006: Data Assimilation as Synchronization of Truth and Model: Experiments with the Three-Variable Lorenz System. *J. Atmos. Sci.* **63**, 2340-2354.